

Visual Steering Commands for Trade Space Exploration: User-Guided Sampling With Example

Gary Stump

e-mail: gms158@psu.edu

Sara Lego

e-mail: ses244@psu.edu

Mike Yukish

e-mail: may106@psu.edu

Applied Research Laboratory,
Pennsylvania State University,
State College, PA 16804

Timothy W. Simpson¹

Department of Mechanical and Nuclear Engineering,
Pennsylvania State University,
University Park, PA 16802
e-mail: tws8@psu.edu

Joseph A. Donndelinger

Vehicle Development Research Laboratory,
General Motors Research and Development Center,
Warren, MI 48090
e-mail: joe.donndelinger@gm.com

Recent advancements in computing power and speed provide opportunities to revolutionize trade space exploration, particularly for the design of complex systems such as automobiles, aircraft, and spacecraft. In this paper, we introduce three visual steering commands to support trade space exploration and demonstrate their use within a powerful data visualization tool that allows designers to explore multidimensional trade spaces using glyph, 1D and 2D histograms, 2D scatter, scatter matrix, and parallel coordinate plots, linked views, brushing, preference shading, and Pareto frontier display. In particular, we define three user-guided samplers that enable designers to explore (1) the entire design space, (2) near a point of interest, or (3) within a region of high preference. We illustrate these three samplers with a vehicle configuration model that evaluates the technical feasibility of new vehicle concepts. Future research is also discussed.

[DOI: 10.1115/1.3243633]

Keywords: multidimensional data visualization, trade space exploration, visual steering

1 Introduction

With recent advances in computing power and speed, designers can now analyze thousands, if not millions, of design alternatives more cheaply and quickly than ever before using simulation or metamodels [1,2]. These advancements provide new opportunities to revolutionize trade space exploration, particularly for the de-

sign of complex systems (e.g., automobiles, aircraft, and satellites) that consist of multiple, interacting subsystems and components that are designed by engineers from a variety of disciplines. The main challenge when designing such systems lies in resolving the inherent tradeoffs that exist both within and between subsystems and the overall system. For example, an aircraft is composed of the wings, fuselage, engines, and countless other subsystems and components. If we consider the design of the wings, tradeoffs exist between aerodynamics, structures, and controls among others, yet wing designers are also driven by the goal to reduce the weight of the wing to minimize the overall weight of the aircraft.

Design optimization is frequently used to help resolve these tradeoffs; however, it still has many shortcomings and challenges to overcome despite recent advances [3]. Balling [4] has noted that the traditional optimization-based design process of “(1) formulate the design problem, (2) obtain/develop analysis models, and (3) execute an optimization algorithm,” often leaves designers unsatisfied with their results because the problem is usually improperly formulated: “the objectives and constraints used in optimization were not what the owners and stakeholders really wanted...in many cases, people don’t know what they really want until they see some designs.” Similar findings have occurred in other fields [5,6].

There is an emerging paradigm of design exploration whereby designers “shop” for the best solution using visualization tools instead of relying solely on optimization. This *design by shopping* process—introduced by Balling [4]—allows designers to explore the design space first, and then choose an optimal solution from a set of possible designs after “forming realistic expectations of what is possible.” The basic steps to such an approach are shown in Fig. 1. First, a simulation model M is created to analyze the system being designed. In many cases, this model is a “black box,” where the relationships between design inputs X and performance outputs Y are not known, and where X and Y combine to form what we call the trade space $Z=[X:Y]^T$. Experiments are then run to simulate thousands of design alternatives by varying X , and storing the corresponding values of Y for each alternative. Interactive visualization tools are then used to explore the trade space to find the most-preferred point Z^* .

1.1 Review of Related Work. Early work in engineering design sought to support this type of approach concentrated on virtual reality to visualize design alternatives. Spherical mechanism design, for instance, significantly benefited from virtual reality [7–9], as have large-scale manufacturing simulations [10,11], fluid mixing simulations [12], and a variety of other engineering design problems [13]. Many researchers examined effective interface development for virtual environments [12,14–16], but most of these virtual environments do not support trade space exploration since they are typically intended to visualize a single point solution, not to explore the entire trade space. Cloud visualization [17], visual design steering [18,19], and the U.S. Naval Research Laboratory’s visual steering methods [20] provide some exceptions to this. Messac and Chen [21] proposed an interactive visualization method, where the progress of the optimization is visualized—but not steered—throughout the design process, not just at the beginning and end. Likewise, visual design steering and graph morphing [18,19] allow users to stop and redirect the optimization process to improve the solution; however, their visualization capabilities are currently limited to 2D and 3D representations of constraints and objectives. Recent work has investigated visualization of n -dimensional Pareto frontiers [22]. Finally, in data mining and knowledge discovery, existing software applications that offer multidimensional data visualization capabilities for trade space exploration include MINER3D, Spotfire’s DECISIONSITE, XMD-VTOOL, and GGOBI—the capabilities of each are reviewed elsewhere [23].

¹Corresponding author.

Contributed by the Simulation and Visualization Committee of ASME for publication in the JOURNAL OF COMPUTING AND INFORMATION SCIENCE IN ENGINEERING. Manuscript received July 28, 2007; final manuscript received July 26, 2009; published online November 2, 2009. Editor: J. Oliver.

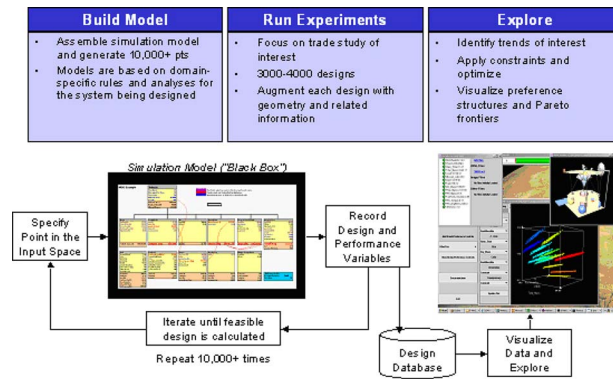


Fig. 1 Typical approach to trade space exploration

1.2 Overview of ARL Trade Space Visualizer. To support trade space exploration, researchers at the Applied Research Laboratory (ARL) and Penn State have developed the ARL trade space visualizer (ATSV) [23,24], a JAVA-based application that displays multidimensional trade spaces using glyph, 1D and 2D histograms, 2D scatter, scatter matrix, and parallel coordinate plots, linked views, and brushing—examples are provided in Sec. 3. The glyph plot capability has been developed using the Visualization Toolkit (VTK), an open source application that supports interactive 3D plots within a JAVA application [25]. VTK provides the capability to view plots in 3D stereoscopic mode, which can then be used within advanced visualization and virtual environments as desired [24]. The ATSV is developed entirely in JAVA, making it cross-platform compatible unlike many commercially available software packages.

In addition to offering these capabilities to visualize multidimensional data, ATSV provides data analysis tools such as brushing [26], linked views [27], Pareto frontier display, preference shading, and data reduction and zooming methods, as shown in Fig. 2, to facilitate trade space exploration. ATSV can import any text-based data file or worksheet, where the rows are the individual design points, the columns are the simulation model's inputs and outputs, and the data is in comma- or tab-delimited format. ATSV can also input trade spaces, where individual designs are tagged with additional files, and when queried, these files (e.g., a 3D solid model of the system) can be displayed, as seen in the middle right of Fig. 1.

In addition to reading in static data sets that have been generated offline, the visual steering commands introduced in Sec. 2 allow users to interactively sample and explore the trade space by directly querying the simulation model to generate data on-the-fly. Section 2 describes how this is achieved, and their use is demonstrated in Sec. 3, with a vehicle configuration model that evaluates the technical feasibility of new vehicle concepts. This model is representative of simulation models commonly used in trade studies, allowing us to demonstrate several important aspects of our software and the proposed visual steering commands. Closing remarks and future research are discussed in Sec. 4.

2 Visual Steering Using Samplers

Before proceeding further, we first define the underlying goal in trade space exploration, as it differs from optimization. We assume the least informative starting point, where the decision makers have no knowledge of their preference on Z or the relationships in M . As they explore the trade space, they will simultaneously form their preference while searching for the most-preferred point in the trade space. We further assume that there exists a utopia point Z^* that the decision makers would choose from Z if they had unlimited ability to explore the trade space completely. Recognizing the finite computing power available and the cost of search time, the goal is to have the decision

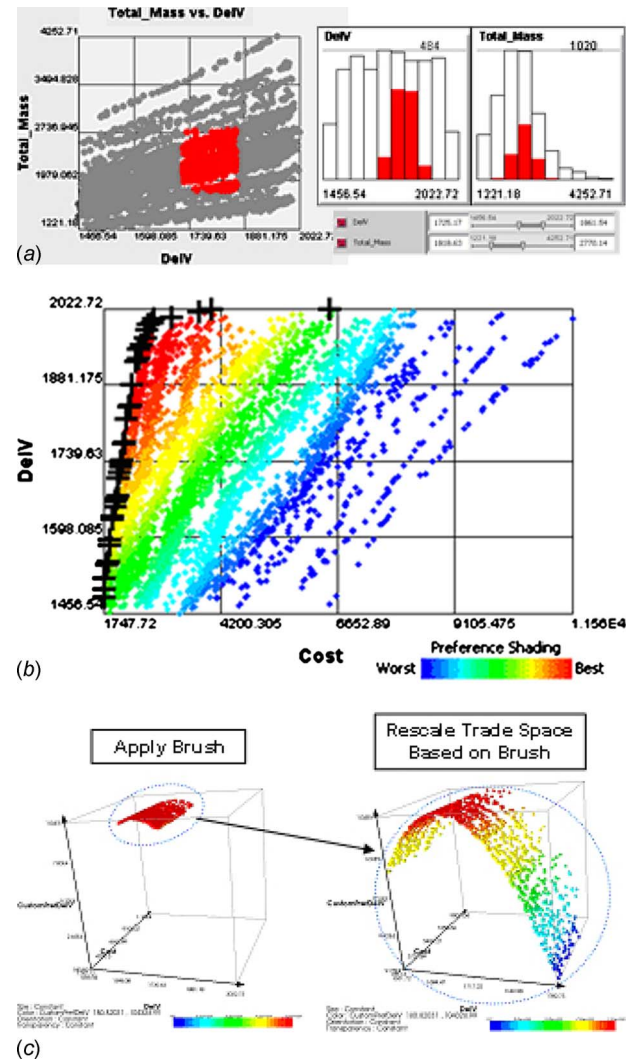


Fig. 2 Examples of data analysis tools in ATSV: (a) linked views that display the same brush settings; (b) Pareto frontier display (+) and preference shading; (c) using brushing to zoom in within the trade space

maker(s) choose a point Z^+ as close as possible to Z^* , while minimizing the time to arrive at the choice of Z^+ . This differs from optimization in that the decision makers' preference is not known a priori, or may change as a result of information gained during the exploration process, sending the search in an entirely different direction.

With this in mind, we are developing visual steering commands that help decision makers form their preference while exploring the trade space (i.e., "shopping") to focus in on regions/points of interest as their preference sharpens. In this paper, visual steering commands are embodied in three user-guided samplers that are created to sample (1) the entire design space, (2) near a point of interest, or (3) within a region of high preference. As echoed in the reinforcement learning literature [28], we have noticed a basic dichotomy when using visual steering commands: (i) those that *explore* the trade space by broadly searching it, and (ii) those that *exploit* knowledge gained during trade space exploration, to guide and narrow the search. Initially, users start out by conducting a broad search, then begin exploring localized regions of the trade space to increase knowledge of the underlying relationships, finally focusing their search in a region potentially containing Z^* . When viewed in this light, the first sampler supports exploration of the entire trade space (case (i)), while the second two exploit

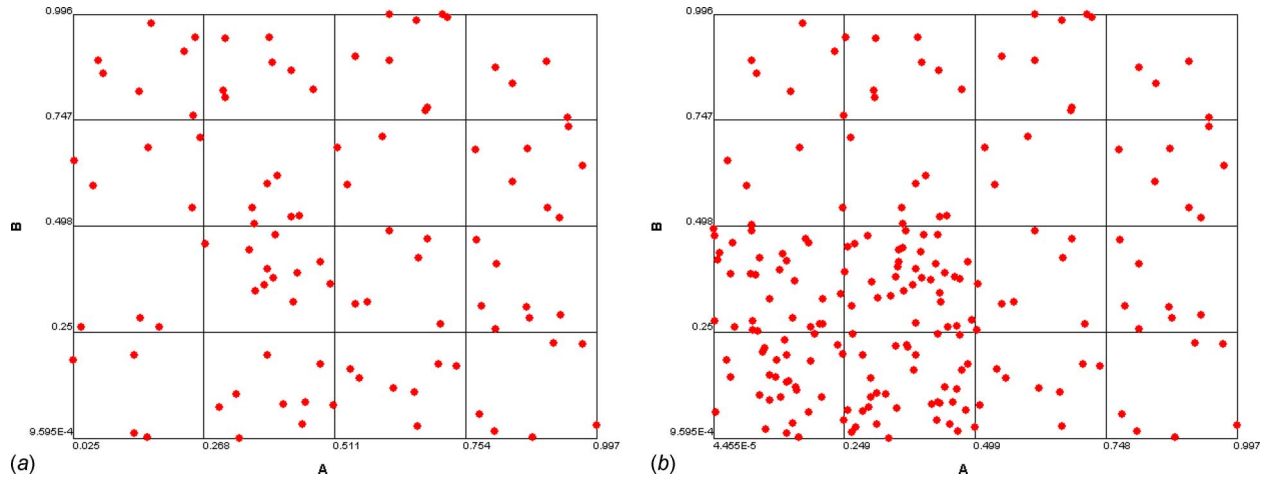


Fig. 3 Design space sampler examples: (a) 100 samples in $A, B \in [0,1]$; (b) 100 new samples in $A, B \in [0,0.5]$

knowledge about points or regions of interest to guide the sampling process (case (ii)). These three samplers can be used together in any combination to alternate between exploring the trade space and exploiting useful information that is obtained during this exploration process. Descriptions of each follow.

- (1) *Design space sampler.* The design space sampler randomly samples over the multidimensional hypercube of X , which is defined by the upper and lower bounds on each input variable. These bounds can be reduced using the brushes in the ATSV to “zoom in” on regions of interest as additional information becomes available. The sampler performs a Monte Carlo simulation on the inputs of the simulation model M , where each input may have a uniform, normal, or triangular distribution. An example is shown in Fig. 3, where Fig. 3(a) shows 100 samples, randomly distributed, based on the original bounds ($0 \leq A, B \leq 1$), while Fig. 3(b) shows the next 100 samples in the reduced region of interest ($0 \leq A, B \leq 0.5$). This sampler is usually the first step in the exploration process in that it initially populates the trade space so that users can start to visualize tradeoffs and trends within the data. While more advanced sampling strategies can be employed (e.g., Latin hypercubes [29], uniform designs [30]), we have found that random sampling is more advantageous when exploring the trade space visually since any structure that occurs in the sample data is an artifact of the model M (potentially valuable information to a decision maker), rather than being induced by the sampling process [31].
- (2) *Attractor sampler.* The attractor sampler—or attractor for short—populates new samples near a user-defined point within the trade space. The attractor is specified in the ATSV interface with a graphical icon \oplus that and is frequently used to try to fill “gaps” in the n -dimensional trade space. Since the user-specified point in the trade space Z can consist of any combination of the inputs X and the outputs Y , and can consist of discrete and continuous variables, we use an evolutionary algorithm, specifically Differential Evolution (DE) [32], to guide the sampling process. The fitness of each new sample point Z_{i_sample} is based on the normalized Euclidean distance from the specified n -dimensional point (i.e., the attractor) $Z_{i_attractor}$

$$\text{Fitness} = \sqrt{\sum_{i=1}^n \left(\frac{Z_{i_sample} - Z_{i_attractor}}{Z_{i_attractor}} \right)^2} \quad (2)$$

The user can specify the population size and number of generations before executing the attractor sampler and can stop the attractor sampler at any time the user desires. As the population evolves, the samples get closer and closer to the attractor icon. As shown in Fig. 4, the attractor is placed in the 2D trade space (see Fig. 4(a)), and additional samples are slowly added near the attractor (see Fig. 4(b)). The default settings for crossover constant (1.0), mutation constant (0.5), and the Best1Bin selection strategy are used within the DE algorithm, and ongoing studies on a variety of test problems—varying in dimensionality, size, nonlinearity, complexity, etc.—are currently being performed to tune DE for use in ATSV. For higher dimensional trade spaces, parallel coordinates is most effective for locating attractors, as demonstrated in Sec. 3. Finally, we note that we do not discard any points in the early generations, even though many are far away from the user-specified point; likewise, we retain points that may become infeasible after applying constraints via brushes. This is in line with the concept that users are simultaneously forming their preference while searching the trade space. The imposition of additional constraints (e.g., maximum weight of satellite due to launch vehicle restriction, maximum beam of a ship) could eliminate all points near an attractor, leaving the trade space unpopulated if all points were not maintained and stored.

- (3) *Preference-based sampler.* The third sampler is a preference-based sampler that populates the trade space in regions that perform well with respect to a user-defined preference function. New sample points are generated using the DE algorithm, but the fitness of each sample point is defined by the user’s preference structure, expressed currently as a linear weighting over Z [23]

$$\text{Fitness} = \sum_{j=1}^{n_{\text{pref}}} w_j Z_j \quad (3)$$

where n_{pref} is the number of trade space variables Z_j to which the user has assigned a preference weight w_j . As shown earlier in Fig. 2(b), ATSV already supports the ability to display preference structures, and this preference structure is retrieved directly from ATSV, and used in the evolutionary algorithm.

An example of the preference-based sampler is shown in Fig. 5. Starting with 100 sample points (see Fig. 5(a)), the preference-based sampler is executed for seven generations using a population size of 25 to obtain the results shown in Fig. 5(b). Note the

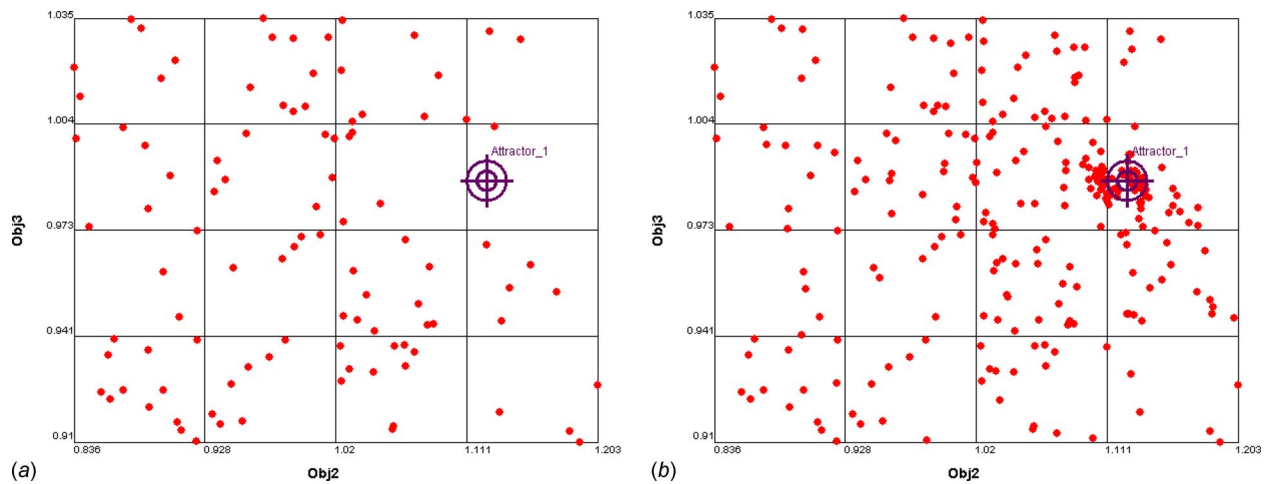


Fig. 4 Generating new samples near a point of interest with the attractor: (a) original 100 sample points; (b) sample points move toward attractor

concentration of points increases in the direction of preference, namely, the upper right hand corner of the plot, which has been specified using the brush controls (see Fig. 5(b)) to maximize both objectives with equal weighting. As with the point sampler, we are currently performing studies to finetune the underlying algorithm, based on the overall problem size and its complexity, as well as the number of trade space variables on which the user has specified a preference.

Implementation of these three samplers occurs through an exploration engine that mediates between ATSV and the simulation model M , as shown in Fig. 6. The exploration engine is written in JAVA and is a subpackage to ATSV. To interface with this exploration engine, ATSV was modified to:

- add user controls (e.g., \oplus) to specify the samplers in ATSV,
- create objects that embody these steering commands, which are passed from ATSV to the exploration engine, and
- allow for continuous display of new points as the exploration engine generates them.

The exploration engine can interface with simulation models coded in JAVA, C, C++, etc. that are executable in “batch mode,” as well as simulation models developed in Excel or MATLAB.

While the implementation of the exploration engine and samplers has been described within the context of ATSV, they are readily applicable to any visualization software that supports trade space exploration. An example application follows.

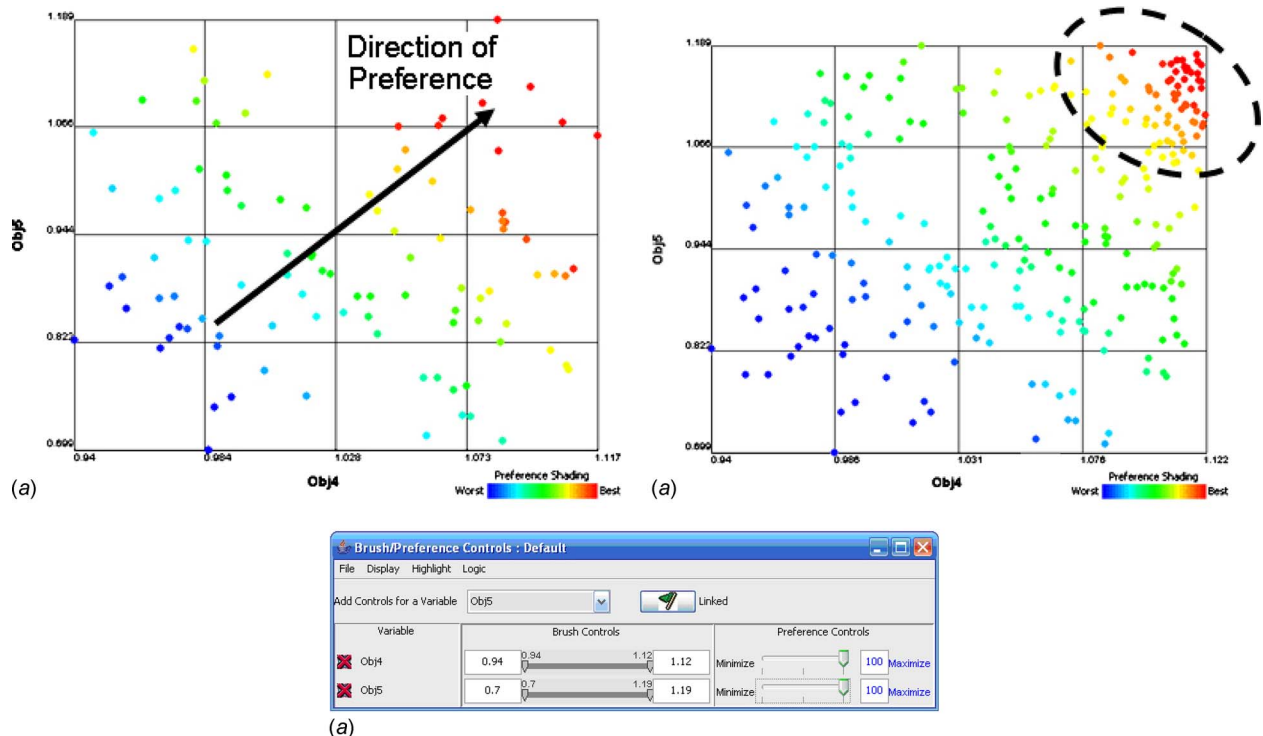


Fig. 5 Preference-based sampler example: (a) original 100 sample points with preference shading; (b) samples increase in direction of preference; (c) brush settings indicating preference structure

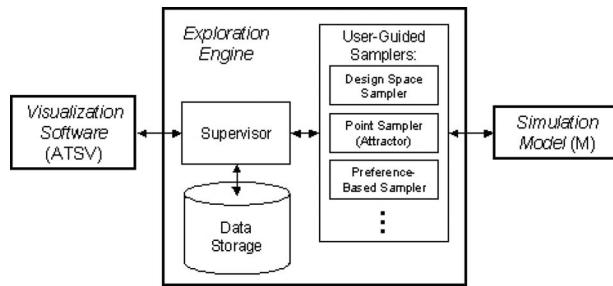


Fig. 6 Exploration engine—system architecture

3 Vehicle Design Example

3.1 Problem Description. To demonstrate the use of visual steering commands and the three samplers for trade space exploration, we linked the exploration engine with a simulation model derived from an existing vehicle configuration model [33,34]. The model includes five measures of performance—acceleration, fuel economy, and measures of interior accommodation—and 11 high-level vehicle design parameters, including ten continuous variables that define overall exterior dimensions and positions of the occupants, and one discrete variable that specifies the vehicle's powertrain. We draw a distinction from previous work with this model in that we do not restrict ourselves to points located in the technical feasibility model (TFM), i.e., the Pareto frontier generated from the vehicle configuration model (see Ref. [35] for a detailed discussion on the TFM). Instead, we compute a constraint function ConVio, which measures the total violation of all constraints in the model, such that feasible points have zero constraint violation, and infeasible ones have a nonzero value for ConVio. This enables us to explore a broader range of design points as we search for overall trends in the data and underlying simulation.

Table 1 summarizes the problem definition that is used for this example. We normalize the bounds on the ten continuous design variables to [0,1], and scale the objectives against the baseline model—defined as the point $Y=(1,1,1,1,1,1)$ with ConVio = 0—to protect the proprietary nature of the data. The design variable H defines the powertrain and can take one of six options [1–6]. Finally, the preference for each objective is indicated in the

Table 1 Vehicle problem definition

Model Inputs		
Variable	Lower bound	Upper bound
A	0	1
B	0	1
C	0	1
D	0	1
E	0	1
F	0	1
G	0	1
H	1,2,3,4,5, or 6	
I	0	1
J	0	1
K	0	1
Model outputs		
ConVio	0 → feasible	>0 → infeasible
Mass	Baseline=1	Defines weight class
Obj1	Baseline=1	Smaller is better
Obj2	Baseline=1	Larger is better
Obj3	Baseline=1	Larger is better
Obj4	Baseline=1	Larger is better
Obj5	Baseline=1	Larger is better

table, and we only want Obj1 to be smaller than the baseline value—larger is better for the other four objectives. While stating these very general preferences beforehand may seem counterintuitive to trade space exploration (i.e., if the direction of preference is known for each objective, then why not just use a multiobjective genetic algorithm to find the best design?), the end goal is to determine the best point Z^+ in the trade space, and to do this, we would need, for example, to specify weights for each objective if we were to use a weighted-sum method to find the best point. The problem arises in that we have no idea what weights to specify for each objective at this stage of the design process—and any weights that we did specify would invariably led to a design that, in all likelihood, was not what we intended to design (see the discussions in Refs. [3,36]). We could use a multiobjective genetic algorithm to determine the Pareto frontier, as was done in earlier work [35]; however, that gives us a set of nondominated designs, not a single point, and this is where trade space exploration can help by allowing designers to shop for the best design. An example of such shopping process follows.

3.2 Visual Steering to Design a New Vehicle. Following the scenario described in Ref. [35], we begin with 78 different vehicle configurations and the goal of finding a new configuration that improves on all objectives simultaneously compared with the baseline model, if possible. After learning how to use ATSV, the visual steering engineer (VIE) starts by looking at a histogram of the initial 78 points (see Fig. 7(a)) to see how well they cover the design space $X=\{A, B, \dots, K\}$. The VIE determines that regions of several design variables (e.g., the middle half of A and E , the upper 3/4 of C , all but the highest values of G , etc.) are under-sampled. The bounds over which A – K are sampled are adjusted, and 22 new sample points are generated using the design space sampler to yield a total of 100 points. The updated histogram in Fig. 7(b) shows how the sampler has added new points in these specific ranges. Upon looking at this figure, the VIE decides that these 100 sample points are sufficient to start exploring the trade space.

Next, the VIE plots a scatter matrix of the 100 points, which is shown in the Appendix, and finds five interesting trends in the plot as follows.

1. Obs. No. 1: As A increases, Mass increases, and as Mass increases, Obj1 increases within each of the discrete levels that appear. These discrete levels appear to correlate with H , the variable related to powertrain selection, and the levels of H also correlate with Obj2, leading to an interesting relationship between Obj1 and Obj2.
2. Obs. No. 2: Low and high values of A tend to violate the constraints (i.e., high values of ConVio) much more than the middle of the range.
3. Obs. No. 3: B is positively correlated with Obj5.
4. Obs. No. 4: J is positively correlated with Obj3.
5. Obs. No. 5: K is positively correlated with Obj4.

Based on these observations, the VIE realizes that B , J , and K can be used to improve Obj5, Obj3, and Obj4, respectively; so, a 2D scatter plot of Obj1 versus Obj2 is created to investigate this relationship further (see Fig. 8(a)). The VIE also assigns color to ConVio in this 2D plot to ascertain the level of feasibility of these points, which has largely been ignored until now. Given the preference for low values of Obj1 and high values of Obj2, the VIE places the $\oplus_{\text{Attractor}_1}$ in the upper left corner of the plot, and executes the attractor sampler for six generations with a population size of 25, using the default settings for crossover, mutation, and selection mentioned in Sec. 2. Figure 8(b) shows the updated plot with these new samples.

The VIE is pleased to see more points clustered near the attractor, as well as the higher concentration of feasible points (in dark blue), but is now starting to thinking about maximizing Obj3—

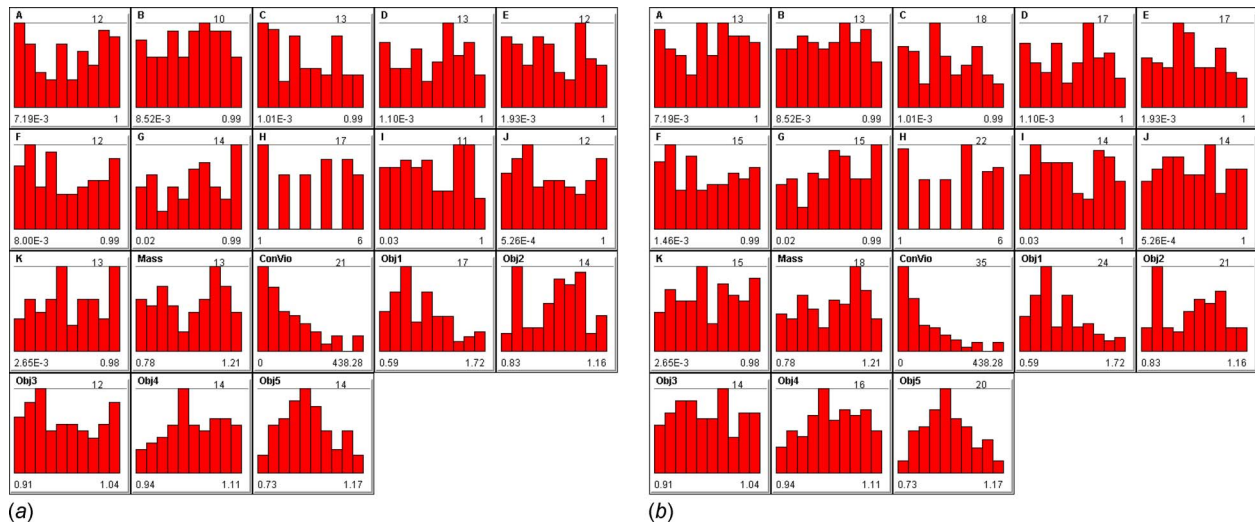


Fig. 7 Use of design space sampler to achieve more uniform sample distributions: (a) histogram of 78 initial vehicle configurations; (b) histograms updated with 22 new samples

Obj5. The VIE places a second attractor in the upper regions of Obj3, Obj4, and Obj5, using a parallel coordinates plot, which easily allows attractors to be placed in more than two dimensions. This second attractor is used to execute a second attractor sampler that runs for six generations, each having a population size of 25, and the results are shown in Fig. 9. The plot is color coded based on ConVio, and the VIE quickly notes the “bands” of feasible (dark blue) points that appear after running the second attractor. Unfortunately, the band originating in the middle of Obj1 and the bands in the lower and middle regions of Obj2 are not helpful, given the preference to minimize Obj1 and maximize Obj2. We note that the dashed oval has been added to highlight the location of $\oplus_{\text{Attractor}_2}$, and the location of $\oplus_{\text{Attractor}_1}$ is also shown.

To gain better insight into the nature of these bands of solutions, the VIE uses a glyph plot to visualize all five objectives at once, in a more familiar 3D representation. In doing so, the VIE takes advantage of the strong correlation between Obj3 and Obj5 (seen in Fig. 9 and the Appendix), plots Obj1, Obj2, and Obj3 on the x, y, and z axes, and assigns Obj4 to size where larger is better. ConVio is assigned to transparency—feasible points are more transparent—and the VIE specifies his preference on each objective using the brush/preference controls so that color in the result-

ing glyph plot indicates preference (see Fig. 10(a)).

The VIE is still worried that many points appear to be infeasible (i.e., are highly transparent in Fig. 10(a)), and a preference-based sampler is executed to try to obtain more feasible points in the region of high preference (red points as indicated by the color scale). This sampler is executed to generate another 150 points (six generations each with a population of 25), and the resulting plot that includes these new samples is shown in Fig. 10(b). The VIE is pleased to see a higher percentage of feasible points and several more points in the preferred region.

The final step in the VIE’s shopping process is to use the brush controls to screen out points that are worse than the baseline design. It turns out that there are no feasible points where all of the objectives are improved as desired, and the VIE must make a tradeoff between which objectives to improve and which to compromise. The VIE learns from brushing the data that it is relatively easy to improve Obj3, Obj4, and Obj5 (by increasing the values for *B*, *J*, and *K* as noted earlier), and that the real tradeoff occurs between Obj1 and Obj2. Using the brush control settings shown in Fig. 11(a), the VIE reduces the points to the few shown in Fig. 11(b), from which a promising option is selected: This design sacrifices only 2% in Obj1 and Obj2 for a gain of 3%, 9%, and

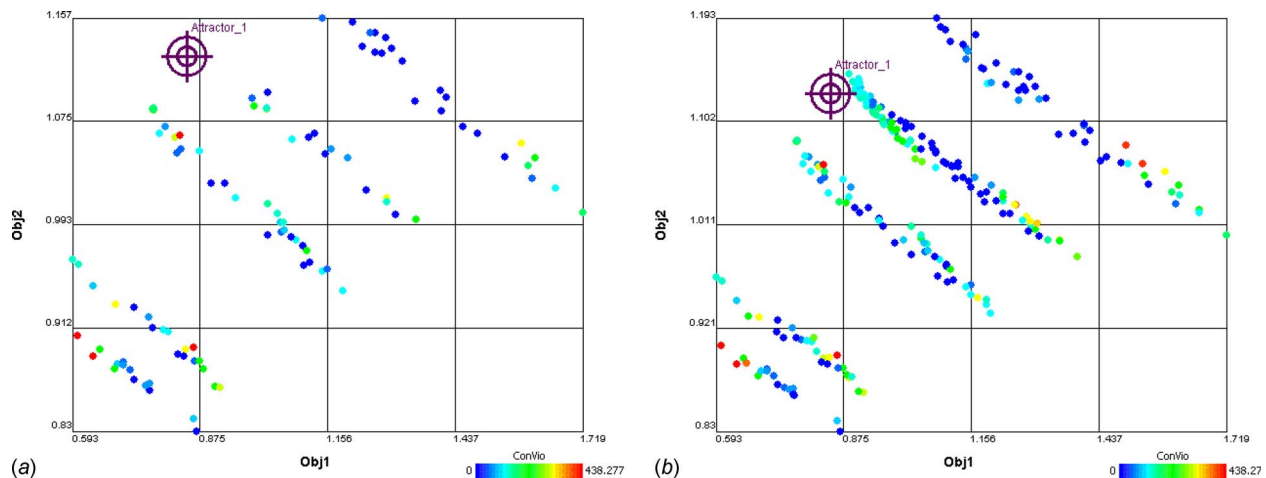


Fig. 8 2D scatter plot of Obj1 versus Obj2: (a) placement of $\oplus_{\text{Attractor}_1}$ in 2-D scatter plot; (b) updated scatter plot after using point sampler

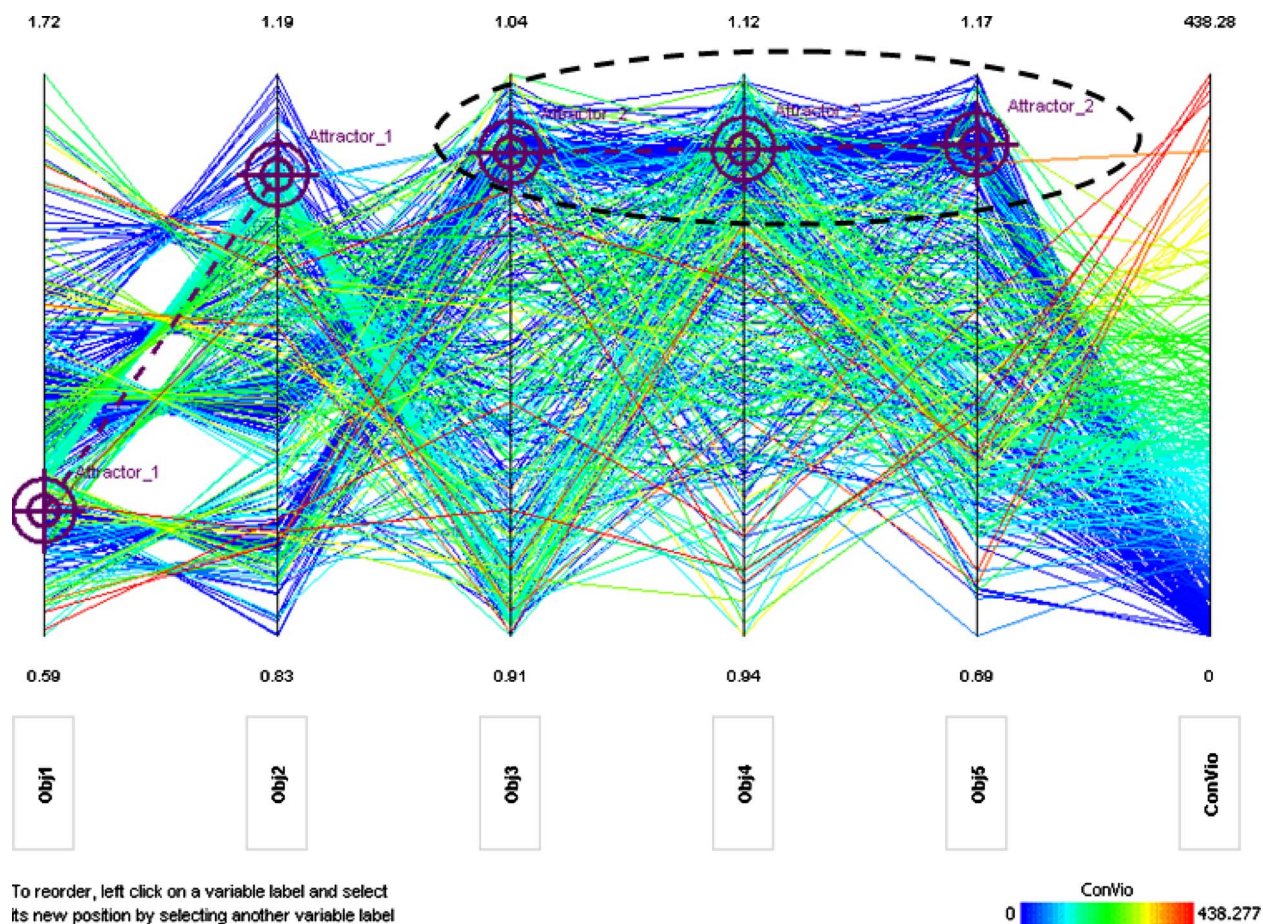


Fig. 9 Parallel coordinates plot showing Obj1–Obj5 and ConVio (with ConVio=color)

13% in Obj3, Obj4, and Obj5, respectively. When creating this final glyph, the VIE added isosurfaces to the plot to indicate different weight classes based on the vehicle mass, and observed that the final options fell within one of three weight classes. This information will be useful as different configurations and powertrain

options are considered during detail design.

In reality, the shopping example could have proceeded in numerous ways using countless different plots and charts. Experienced users might prefer to brush the data against the baseline vehicle, and start shopping from there, whereas novice users

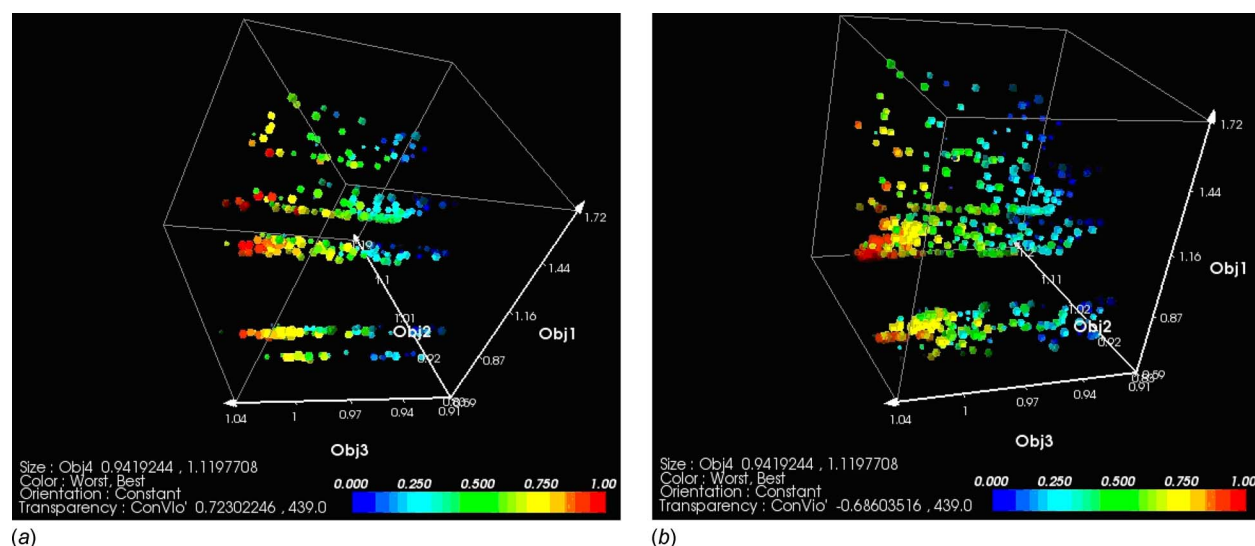
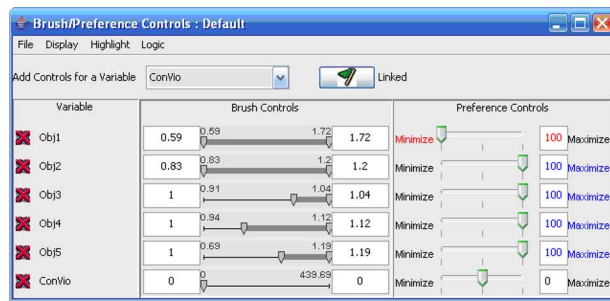
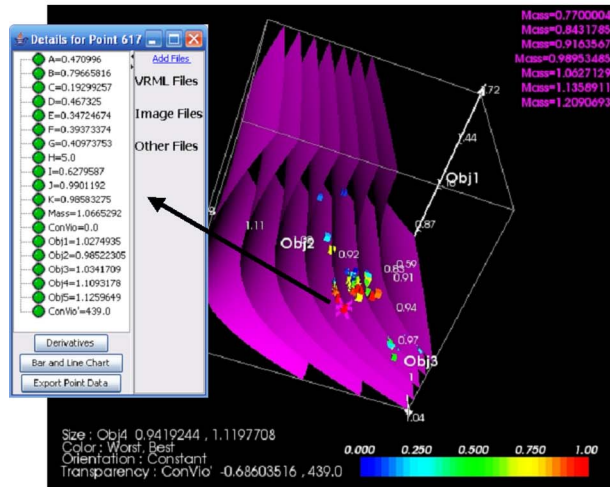


Fig. 10 Plot of Obj1–Obj3 with Obj4=size, ConVio=transparency, and Preference shading=color: (a) glyph plot of current designs; (b) glyph plot after preference-based sampling



(a)



(b)

Fig. 11 Selection of final vehicle design point: (a) brush/preference controls for making final selection; (b) resulting glyph plot including mass contours

might want to gain insight into the model during the shopping process while also locating the best design. The story, as we have told it, is representative of what we have observed at this particular company for this particular problem; however, we have clearly articulated each step to demonstrate the visual steering commands, in general, and the three samplers, in particular. Future work will investigate any common processes or search strategies, as outlined in the Sec. 4.

4 Closing Remarks and Future Work

In this paper, we have discussed the need for trade space exploration, and introduced a set of visual steering commands to support the process. To enable these commands, we have defined and implemented three user-guided samplers that enable designers to explore (1) the entire design space, (2) near a point of interest using an attractor, or (3) within a region of high preference, as specified by the user. Finally, we demonstrated these visual steering commands to guide the trade space exploration process using a vehicle configuration model. Within this context, we highlighted the features and capabilities of our powerful new data visualization tool, ATSV, which offers multidimensional viewing capabilities, including any combination of glyph, 1D and 2D histograms, 2D scatter, scatter matrix, and parallel coordinate plots, linked views, brushing, preference shading, and Pareto frontier display.

In addition to the ongoing work discussed in this paper, there

are a variety of promising avenues for future research. For instance, the three samplers presented in this paper represent our initial attempts at creating visual steering commands to guide trade space exploration. We envision a variety of additional samplers such as a Pareto sampler, most informative sampler, most uncertain sampler, etc., which support different aspects of the shopping process. We also envision a “repeller” that works in the exact opposite way as the attractor does, i.e., generates points that are NOT like the user-specified point. We have already encountered several applications working with companies where this type of sampler would be useful. We are also in the process of interviewing designers, engineers, and practitioners to learn more about how they use visualization to support decision making and the capabilities that they wished they had for trade space exploration. One example is the capability to visualize the corresponding physical geometry for any selected point in the trade space, and we are investigating ways to link and render 3D parametric models within ATSV to support this.

Two underlying assumptions when using visual steering commands are that (i) a simulation model M is available to query, and (ii) the analyses are not computationally expensive, and can be executed in real-time, or sufficiently quickly, as the case may be. For case (i), the ATSV was initially designed to work with static data sets, and can incorporate empirical data, as long as it is in tabular format. If an underlying model does not exist, then regression techniques could be used to create response surfaces [37], for instance, based on the empirical data, which could then be explored within ATSV. Likewise, for case (ii), a variety of metamodeling techniques exist that can be used to construct inexpensive surrogates of any computationally expensive analyses [37], and these could be queried directly by the *exploration engine* when executing any of these samplers since it is the runtime of the simulation model that constrains their use. The visual steering commands could then be used to infer regions of interest, which can drive toward regions in the trade space, wherein the computationally expensive analyses are executed next. We have found that the simplified models typically used for trade studies such as demonstrated here are not computationally expensive, and lend themselves well to be used in conjunction with the proposed visual steering commands.

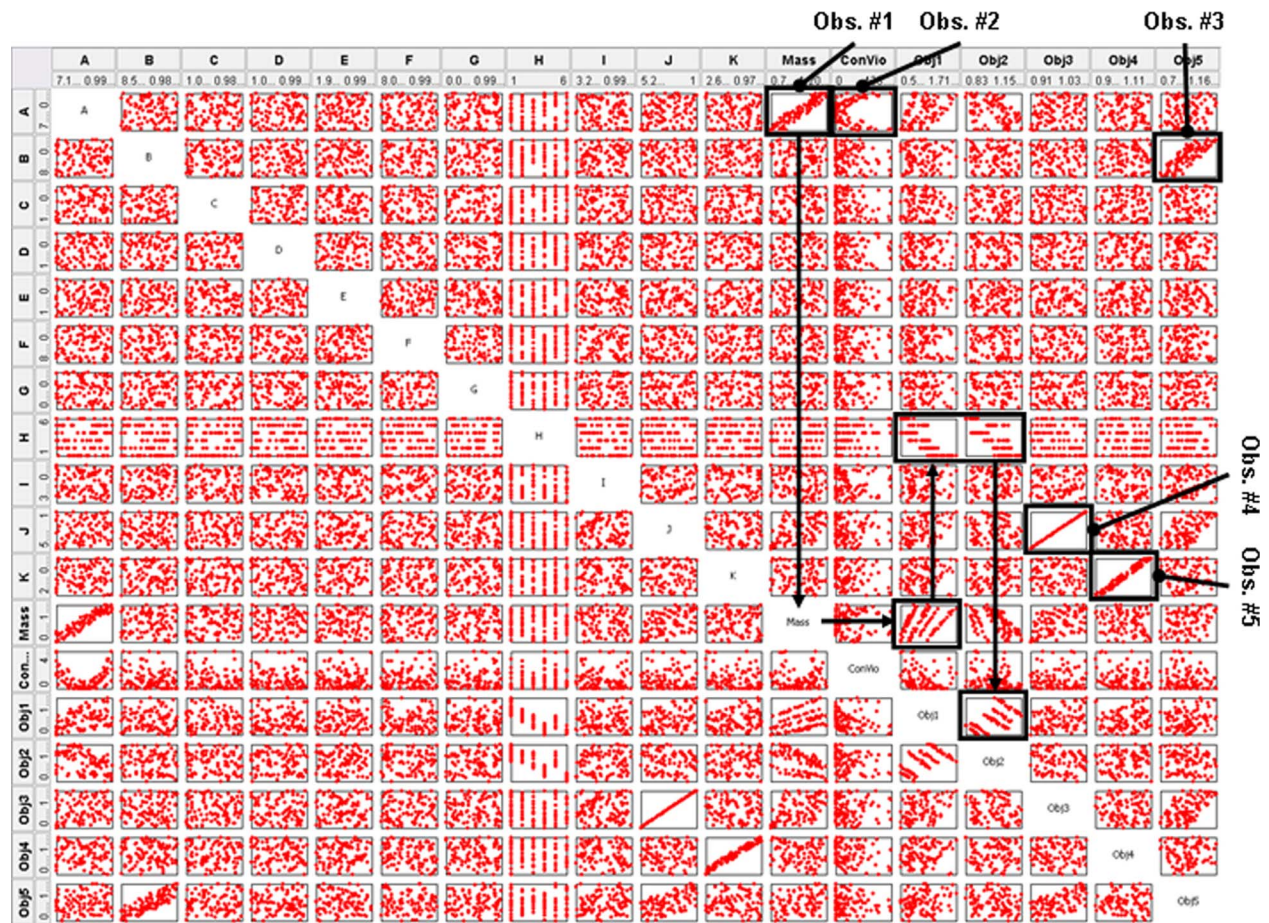
Finally, there is the issue of training users in how to use (a) ATSV and (b) the visual steering commands and samplers. Given the variety of data visualization capabilities within ATSV, users can quickly become overloaded, especially when considering high dimensional trade spaces. We plan to begin developing videos and training protocols to improve proficiency with ATSV, and instruct users on how to best utilize its capabilities and visual steering commands. Meanwhile, we have observed on several occasions that users already familiar with ATSV can quickly learn how to utilize the visual steering commands and samplers in the ATSV interface, which demonstrates to us that they are fairly intuitive to use.

Acknowledgment

We thank Scott Ferguson for his assistance with the vehicle configuration model. This work has been supported in part by the National Science Foundation under Grant No. CMMI-0620948. Any opinions, findings, and conclusions or recommendations presented in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Appendix

A scatter matrix plot of 100 samples points to identify trends.



References

- [1] Wang, G. G., and Shan, S., 2007, "Review of Metamodeling Techniques in Support of Engineering Design Optimization," *ASME J. Mech. Des.*, **129**(4), pp. 370–380.
- [2] Venkataraman, S., and Haftka, R. T., 2004, "Structural Optimization Complexity: What Has Moore's Law Done for Us?," *Struct. Multidiscip. Optim.*, **28**(6), pp. 375–387.
- [3] Papalambros, P., 2002, "The Optimization Paradigm in Engineering Design: Promises and Challenges," *Comput.-Aided Des.*, **34**(12), pp. 939–951.
- [4] Balling, R., 1999, "Design by Shopping: A New Paradigm?," *Proceedings of the Third World Congress of Structural and Multidisciplinary Optimization (WCSMO-3)*, Buffalo, NY, University at Buffalo, pp. 295–297.
- [5] Shanteau, J., 1992, "Competence in Experts: The Role of Task Characteristics," *Org. Behav. Hum. Decis. Process.*, **53**(2), pp. 252–266.
- [6] Wilson, T. D., and Schooler, J. W., 1991, "Thinking Too Much: Introspection Can Reduce the Quality of Preferences and Decisions," *J. Pers. Soc. Psychol.*, **60**(2), pp. 181–192.
- [7] Evans, P. T., Vance, J. M., and Dark, V. J., 1999, "Assessing the Effectiveness of Traditional and Virtual Reality Interfaces in Spherical Mechanism Design," *ASME J. Mech. Des.*, **121**(4), pp. 507–514.
- [8] Furlong, T. J., Vance, J. M., and Larochelle, P. M., 1999, "Spherical Mechanism Synthesis in Virtual Reality," *ASME J. Mech. Des.*, **121**(4), pp. 515–520.
- [9] Kihong, J. N., Vance, J. M., and Larochelle, P. M., 2002, "Spatial Mechanism Design in Virtual Reality With Networking," *ASME J. Mech. Des.*, **124**(3), pp. 435–440.
- [10] Kelsick, J., Vance, J. M., Buhr, L., and Moller, C., 2003, "Discrete Event Simulation Implemented in a Virtual Environment," *ASME J. Mech. Des.*, **125**(3), pp. 428–433.
- [11] Kesavadas, T., and Sudhir, A., 2000, "Computational Steering in Simulation of Manufacturing Systems," *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, CA, IEEE, pp. 2654–2658.
- [12] Duncan, T. J., and Vance, J. M., 2007, "Development of a Virtual Environment for Interactive Interrogation of Computational Mixing Data," *ASME J. Mech. Des.*, **129**(3), pp. 361–367.
- [13] Jayaram, S., Vance, J. M., Gadh, R., Jayaram, U., and Srinivasan, H., 2001, "Assessment of VR Technology and Its Applications to Engineering Problems," *ASME J. Comput. Inf. Sci. Eng.*, **1**(1), pp. 72–83.
- [14] Mulder, J. D., van Liere, R., and van Wijk, J. J., 1998, "Computational Steering in the CAVE," *FGCS, Future Gener. Comput. Syst.*, **14**(3–4), pp. 199–207.
- [15] Volkov, S., and Vance, J. M., 2001, "Effectiveness of Haptic Sensation for the Evaluation of Virtual Prototypes," *ASME J. Comput. Inf. Sci. Eng.*, **1**(2), pp. 123–128.
- [16] Balijepalli, A., and Kesavadas, T., 2004, "Value-Addition of Haptics in Operator Training for Complex Machining Tasks," *ASME J. Comput. Inf. Sci. Eng.*, **4**(2), pp. 91–97.
- [17] Eddy, J., and Lewis, K., 2002, "Visualization of Multi-Dimensional Design and Optimization Data Using Cloud Visualization," *ASME Paper No. DETC02/DAC-02006*.
- [18] Winer, E. H., and Bloebaum, C. L., 2001, "Visual Design Steering for Optimization Solution Improvement," *Struct. Multidiscip. Optim.*, **22**(3), pp. 219–229.
- [19] Winer, E. H., and Bloebaum, C. L., 2002, "Development of Visual Design Steering as an Aid in Large-Scale Multidisciplinary Design Optimization. Part I: Method Development," *Struct. Multidiscip. Optim.*, **23**(6), pp. 412–424.
- [20] Smith, W. R., Anderson, W. L., Haftel, M. I., Kuo, E., Rosen, M., and Uhlmann, J. K., 1999, "Goal-Oriented Computational Steering," *Proceedings of the SPIE Conference on Enabling Technology for Simulation Science III*, Orlando, FL, SPIE, pp. 250–260.
- [21] Messac, A., and Chen, X., 2000, "Visualizing the Optimization Process in Real-Time Using Physical Programming," *Eng. Optimiz.*, **32**(6), pp. 721–747.
- [22] Agrawal, G., Lewis, K., Chugh, K., Huang, C.-H., Parashar, S., and Bloebaum, C. L., 2004, "Intuitive Visualization of Pareto Frontier for Multi-Objective Optimization in N-Dimensional Performance Space," *Proceedings of the Tenth AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, NY, AIAA Paper No. AIAA-2004-4434.
- [23] Stump, G., Yukish, M., Simpson, T. W., and Harris, E. N., 2003, "Design Space Visualization and Its Application to a Design by Shopping Paradigm," *ASME Paper No. DETC2003/DAC-48785*.
- [24] Stump, G., Yukish, M., and Simpson, T. W., 2004, "The ARL Trade Space Visualizer: An Engineering Decision-Making Tool," *Proceedings of the Tenth AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, NY, AIAA Paper No. AIAA-2004-4568.
- [25] Schroeder, W., Martin, K., and Lorensen, B., 2004, *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*, Kitware, Clifton Park, NY.
- [26] Becker, R. A., and Cleveland, W. S., 1987, "Brushing Scatterplots," *Technometrics*, **29**(2), pp. 127–142.

- [27] Buja, A., McDonald, J. A., Michalak, J., and Stuetzle, W., 1991, "Interactive Data Visualization Using Focusing and Linking," *Proceedings of the IEEE Conference on Visualization '91*, IEEE, New York, pp. 156–163.
- [28] Sutton, R. S., and Barto, A. G., 1998, *Reinforcement Learning: An Introduction*, MIT, Cambridge, MA.
- [29] McKay, M. D., Beckman, R. J., and Conover, W. J., 1979, "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, **21**(2), pp. 239–245.
- [30] Fang, K.-T., Lin, D. K. J., Winker, P., and Zhang, Y., 2000, "Uniform Design: Theory and Application," *Technometrics*, **42**, pp. 237–248.
- [31] Yukish, M. A., 2005, "Research Topics in Trade Space Exploration," *Proceedings of the IEEE Aerospace Conference, Big Sky, MT*, IEEE, pp. 4220–4226.
- [32] Price, K., Storn, R., and Lampinen, J., 2005, *Differential Evolution—A Practical Approach to Global Optimization*, Springer-Verlag, Berlin.
- [33] Ferguson, S., Gurnani, A., Donndelinger, J., and Lewis, K., 2005, "A Study of Convergence and Mapping in Multiobjective Optimization Problems," *ASME Paper No. DETC2005/CIE-84852*.
- [34] Ferguson, S., Gurnani, A., Donndelinger, J., and Lewis, K., 2005, "An Approach to Feasibility Assessment in Preliminary Design," *ASME Paper No. DETC2005/CIE-84853*.
- [35] Donndelinger, J., Ferguson, S., and Lewis, K., 2006, "Exploring Mass Trade-Offs in Preliminary Vehicle Design Using Pareto Sets," *Proceedings of the 11th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Portsmouth, VA, AIAA Paper No. AIAA-2006-7056.
- [36] Balling, R. J., Taber, J. T., Brown, M. R., and Day, K., 1999, "Multiobjective Urban Planning Using Genetic Algorithm," *J. Urban Plann. Dev.*, **125**(2), pp. 86–99.
- [37] Simpson, T. W., Peplinski, J., Koch, P. N., and Allen, J. K., 2001, "Metamodels for Computer-Based Engineering Design: Survey and Recommendations," *Eng. Comput.*, **17**(2), pp. 129–150.