Early-Stage Uncertainty: Effects of Robust Convex Optimization on Design Exploration

Proceedings of the ASME 2020
International Design Engineering Technical Conferences
and Computers and Information in Engineering Conference
IDETC/CIE2020
August 17-19, 2020, Virtual, Online

DETC2020-22626

EARLY-STAGE UNCERTAINTY: EFFECTS OF ROBUST CONVEX OPTIMIZATION ON DESIGN EXPLORATION

Priya P. Pillai
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA, USA

Edward Burnell
Xiqing Wang
Maria C. Yang
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA, USA

ABSTRACT

Engineers design for an inherently uncertain world. In the early stages of design processes, they commonly account for such uncertainty either by manually choosing a specific worst-case and multiplying uncertain parameters with safety factors or by using Monte Carlo simulations to estimate the probabilistic boundaries in which their design is feasible. The safety factors of this first practice are determined by industry and organizational standards, providing a limited account of uncertainty; the second practice is time intensive, requiring the development of separate testing infrastructure. In theory, robust optimization provides an alternative, allowing set based conceptualizations of uncertainty to be represented during model development as optimizable design parameters. How these theoretical benefits translate to design practice has not previously been studied. In this work, we analyzed present use of geometric programs as design models in the aerospace industry to determine the current state-of-the-art, then conducted a human-subjects experiment to investigate how various mathematical representations of uncertainty affect design space exploration. We found that robust optimization led to far more efficient explorations of possible designs with only small differences in an experimental participant's understanding of their model. Specifically, the Pareto frontier of a typical participant using robust optimization left less performance "on the table" across various levels of risk than the very best frontiers of participants using industry-standard practices.

Keywords: robust design, uncertainty modeling, design optimization, multiobjective optimization.

1 INTRODUCTION

Engineering designers use complex computational models to represent a variety of problems, despite their awareness that the results will not be perfectly recreatable in the physical world. Even if a model were able to represent a specific problem perfectly, environmental conditions and physical realities are rarely stable or knowable; for example, an engineer may declare the density of a metal as a particular value, but in manufacturing the metal supplied will vary from supplier to supplier and day to day. Beyond material quantities, such uncertainty is also inevitable for environmental conditions, assembly quality, and many other important components of performance. Accounting for such uncertainty is therefore a necessity which designers often represent through the manual implementation of conservative heuristics.

Convex Geometric Programs (GPs), sets of algebraic constraints globally optimizable for a specific cost function, are capable of representing a variety of complex systems. Historically, the inaccessibility of software used to create and solve GPs has restricted their use in engineering design. The Python package GPkit provides a familiar and clear syntax for geometric programs, reducing this barrier to entry. [1]. Through GPkit, several engineering design firms have adopted GPs for regular use in their processes, typically to validate the feasibility of innovative conceptual designs.

At present, GPkit models (along with most other design models) do not provide interfaces specifically for the representation of uncertainty. Designers instead set some parameters' values to a "reasonable worst case", often via multiplication by a blanket "safety factor". Robust optimization aims to address this by allowing specified uncertainties to be set on parameters, then optimizing for the best worst-case performance under a given uncertainty set [2]. This method provides more mathematical guarantees than safety factors do and is more directly translatable to a simulation environment.

How much these mathematical details affect designers and design practice is unclear. The marginal improvement in design quality may or may not be worth the effort of changing designer's conceptualizations of their model. However, we argue that robust optimization's potential benefits come not only from its underlying mathematics, but also from the novel "questions" it lets designers ask of their models. When uncertainty is explicitly defined in robust GPs, it can be optimized for as if it were any other variable. This provides a dynamic understanding of uncertainty, encouraging discussions of robustness earlier in a design process. This study seeks to explore ways in which robust optimization can affect the practice of creating designs, and provides evidence that robust GPs improve design space exploration, increasing designs' quality, quantity, and coverage relative to an underlying Pareto frontier of optimal tradeoffs.

1.1 Research Questions

Previous work has shown that robust optimization provides a mathematically rigorous method of accounting for uncertainty [3,4]. However, its effects on the questions designers ask of their models has not yet been analyzed. In this study, we ask the following questions:

RQ1 How do designers conceptualize uncertainty? How do particular conceptualizations change their comfort with robust optimization?

RQ2 How do different mathematical formulations of uncertainty, as represented in a design model, affect designers' explorations of possible designs?

RQ3 What design processes do robust optimization tools alter or automate?

Our study had two stages. The first, a series of practitioner field interviews, was used to guide the design of the second, a human-subjects experiment in a controlled environment. We address RQ1 by summarizing how current users of GPkit account for uncertainty in their design processes and looking at how experimental participants used robust optimization to account for uncertainty. RQ2 is addressed by analysis of the quality and spread of experimental participants' solutions. RQ3 is touched on in comparisons between processes for uncertainty accounting described in interviews and those seen experimentally, but we anticipate its full investigation to also require field studies of how robust optimization affects organizational processes.

2 BACKGROUND

A substantial amount of research has been conducted on software tools for design, analysis, and robust optimization, but the development of particular tools is not our focus. Rather, we are interested in how designers use these tools and how the choice, application, and integration of these tools can impact design process exploration. The set of tools we use varies in their handling of uncertainty and robustness. To better specify this variety, we define uncertainty as variables listed as a fixed constant in our model having instead a set of possible values. Robustness is the ability of the design to still function with small perturbations of these fixed variables; the larger a perturbation that can be handled, the more robust the design is.

2.1 Frameworks for Early Stage Design

Many frameworks exist for early stage design processes for products and engineered systems, including Pahl and Beitz' systematic approach to engineering design and Ulrich, et al.'s widely known process for product design and development [5, 6]. Underpinning both approaches is the notion of a design specification and/or initial prototype created by an engineering and design team. The initial prototypes being considered in this study are Python codes using the GPkit library [1]. The current design specification of these models does not include a method of accounting for uncertainty; we will refer to the additional design specification of uncertainty as the conceptualization of uncertainty within the model.

2.2 Design Models

Human participants in engineering organizations use software "design models" to enumerate parameters of their designs and implement interactions amongst these parameters. Design models are often made from materials like parameterized CAD assemblies (to construct a shape from geometric constraints) [7,8], spreadsheets (to calculate performance) [9,10], and "mathematical programs" (to take in a desired performance and put out a design that achieves it) [11].

Design models serve as loci for understanding what will be built, while encoding (and sometimes concealing) decisions on why [12]. This makes them an important arena for intra-organizational design politics, but just how participants' perspectives clash and coalesce around these models depends also on the motif they are part of [7, 13]. Design models express their agency both by shaping the motif and, within a motif, by determining their outsiders and insiders, spectators and maintainers, and formal and informal power structures [7, 14].

2.3 Design Tools and the Designer

Software tools, most notably CAD, are essential to design and production, and a number of studies have considered the impact of these tools on early stage designs. In the exploratory phases of design, studies with practicing engineers and student designers have observed that the use of CAD too early in the design process can have a negative effect on design creativity, known as "premature fixation" [12, 15]. High fidelity digital tools require more time and effort on the part of the designer than lower fidelity tools, making designers more invested in a design and less likely to discard it. This is an observation of not only the design tool, but the way that designers use the tools in practice [16]. Our study takes a similar designer-focused perspective on exploration using a design tool by formulating a constrained but realistic design problem with minimal interface complexity. Our design tool is GPkit, and we investigate the effect of a more detailed but potentially confusing mathematical model of uncertainty on the ability of users to find optimal solutions using this tool. The exact mathematics behind how uncertainty is calculated will be referred to as the formulation of uncertainty.

2.4 Design Optimization and the Designer

An overarching goal of design optimization research is to create tools and systems that can support designers by generating the "best" solutions by searching through the set of all possible solutions, or the design space. The majority of research in design optimization concentrates on the development of better and faster algorithms and strategies, and only limited research has been conducted on how designers themselves reach globally- or locally-optimal solutions, and how this is affected by their tools.

In an early study of how humans deal with coupled problems, Hirschi and Frey compared the time to solve coupled and uncoupled parametric design problems [17]. For uncoupled problems, the time to solve was of the order of $O(n)$ where n is the number of input variables, and increased dramatically to $O(n^{3.4})$ for coupled problems. Notably, coupled problems with more than 4 variables were found to be very difficult and frustrating for the participants. Similarly, human studies by Flager et al. showed that an increase in problem complexity caused a significant decrease in solution quality [18]. A study by McComb et al. showed specifically that more complex 2D trusses led to worse performance [19]. Austin-Breneman et al. found that, despite domain expertise and optimization training, graduate students asked to collaboratively design a simplified satellite had trouble exploring the design space because of the complexity of subsystems and subsystem interactions, and few teams found designs on the Pareto-optimal frontier [20]. In interviews with space system designers, it was found that teams in industry routinely restricted the information shared with each other in ways that made exploration much more difficult both in practice and from the perspective of optimization theory [21]. Yu's study of desalination systems found that software choices could enable novices to explore complex system designs almost as well as experts, with some caveats [22]. Designer satisfaction with rapid prototyping process has been explored by Neeley, et al., who found that designers tended to be more satisfied with design outcomes when given the opportunity to explore more design space initially [23]. Specific questions of how real-time interfaces affect design outcomes were present in the first direct-manipulation CAD software [8], in early studies of the effect of analysis speed on structural design exploration and outcomes [24], and in more recent research on human-computer optimization in circuit-routing [25] and in architectural design [26].

We hope to extend such studies by directly measuring the effects of real-time software decisions and algorithms on design outcomes and process. Previous studies by Barron et al. and Egan et al. [27, 28] have looked at the effects of visualization and search techniques in custom tools that use different visual representations and search strategies than designers may be accustomed to; in contrast, our study uses familiar visual representations and interaction modalities but changes the conceptualization and formulation of the design problem. Since this design problem has two goal parameters, we define "optimality" in terms of the Pareto frontier—a subset of the possible solutions such that each solution on the Pareto frontier is either better in the first goal parameter or the second goal parameter compared to any other solution.

2.5 Geometric Programs

Geometric programs are nonlinear optimization problems of a set of posynomial constraints and a cost function known as the objective. A posynomial is a sum of monomials, where a monomial is a set of variables raised to any positive real power multiplied together with a positive coefficient. Formally, a posynomial $p(x)$ can be defined as

$$p(x) = \sum_{k=1}^{K} c_k \prod_{j=1}^{n} x_j^{a_{j,k}} \tag{1}$$

where x is a vector of all variables, n is the length of x and therefore the number of variables, K is the number of monomials, all c_k are positive real numbers, and all $a_{j,k}$ are real numbers [29].

A geometric program is defined by minimizing a posynomial objective function subject to posynomial constraints that must be less than or equal to some positive value. Geometric programs have the practical feature that, when transformed logarithmically, they become convex, guaranteeing only one local minimum exists-the global minimum. This allows for gradient descent in log-space to always find the globally optimal solution. GPkit serves as a Python interface for geometric program solvers such as MOSEK and cvxopt [30, 31] that allows users to define these objectives and constraints intuitively. It then can solve for the optimal solution and can visualize the structure of the models and the feasible solution space. GPkit has enabled engineering designers who are not experts in mathematical optimization to create, solve, and understand GP models by black-boxing computational details and providing diagramatic representations of the underlying mathematics. If negative c_k values are necessary, a signomial program can be used, which can be optimized via multiple geometric program approximations.

2.6 Robust Convex Optimization

While geometric programs are highly generalizable, they run the risk of being overly specialized solutions relative to the uncertainty that exists. To account for that uncertainty, Robust, an add-on GPkit package, allows for the inclusion of standard deviations on each variable, as well as an overall "Gamma" factor (γ) that scales the amount of uncertainty accounted for, then optimizes the worst point of a region of uncertain parameters. The region can either account for a certain number of standard deviations of each parameter ("rectangular" uncertainty) or of a combination of all parameters ("elliptical" uncertainty). This process is generally known as robust optimization. Work on Robust has shown that the current standard of multiplying each uncertain variable by a margin does not actually take into account the worst combined case mathematically, and that robust optimization is necessary to fully account for uncertainty [3]. While the quantitative case for using Robust has been made, the question of how this affects the overall design process, particularly in the context of design space exploration, has not yet been answered.

3 PRACTITIONER INTERVIEWS

This study was divided into two stages. The first exploratory stage—practitioner interviews—produced qualitative data on Robust adoption's benefits, risks, obstacles, and conditions. From the information gathered in these interviews, we designed the experimental second stage to address the concerns raised and to provide these users with further guidance on how and when to incorporate robust optimization into their existing models.

3.1 Methods

To understand current practices of accounting for uncertainty in design models, we interviewed five GPkit users with a flexible questionnaire focusing on how they accounted for uncertainty within their models. Each of the five interviews lasted for half an hour to an hour and took place off campus, either at the interviewee's place of work or at a public location like a coffee shop. Interviewees varied in the extent of their experience with GPkit, their interactions with GPkit (developers versus designers), and their affiliations (academic versus commercial), though all were in the field of aerospace, where most GPkit models are made. First, we asked about each designer's work to encourage engagement in the conversation and to understand their background. We then explored the workflows of their projects before and after using GPkit, asking them to speak of particular projects to ground their answers. We then asked more targeted questions about uncertainty, looking for specific methods. Finally we asked broadly about inefficiencies they had encountered while modeling, to understand how salient issues surrounding uncertainty are relative to other concerns. Conversations were analyzed using open coding.

TABLE 1. PRACTITIONER DEMOGRAPHICS
Each column represents an interviewed practitioner, each row a trait. An "X" indicates that the practitioner has this trait. "Developer" means they have been involved in GPkit's development process; "Designer" means they have created GPkit models as a part of a longer product development process. "Academic" and "Commercial" refer to the contexts in which the practitioner has worked with GPkit. "Experienced" refers to having multiple years of experience using GPkit.

	1	**2**	**3**	**4**	**5**
Developer		X	X		
Designer	X	X		X	X
Academic	X	X	X	X	
Commercial	X	X			X
Experienced	X	X	X	X	

These interviews were the backbone of our experimental design for the second stage, for we based its guiding questions on the concerns expressed by those interviewed.

3.2 Results

When we asked interviewees how they accounted for uncertainty during conceptual stages of design, we received two responses: either they 1) multiplied uncertain parameters by a margin or safety factor of 20% (considered an industry standard) or 2) did not account for uncertainty at those stages. Some interviewees mentioned checking if their design was robust to small perturbations in environmental conditions via Monte Carlo simulation, but usually as a final check of a model's solution, not during model development. Most interviewees believed they should be accounting for uncertainty, but did not consider it a priority due to a perceived lack of social pressure to do so; if none of their peers were trying to account for uncertainty, why should they? Almost everyone interviewed considered uncertainty quantification an important problem, but also thought of it as intractable and impractical.

Interviewees discussed how safety factors can lead a design to be incorrectly seen as infeasible. One talked in particular about electric airplanes, much of whose mass rests in their battery. Putting a safety factor on total airplane weight increases the amount of battery needed, which increases the total airplane weight; the process converges, but often leaves a design looking impossible. Therefore, instead of weight safety factors, this interviewee accounted for excess weight by making the allowable payload a maximized free variable, even though this makes it more difficult to design for an exact payload.

Deciding on a model's objective function—the parameter it optimizes for—was described as the "single most important choice" of modeling. In robust optimization, uncertainty can be the optimized parameter. This allows for different conceptualizations of a design problem. With the electric aircraft above, instead of calculating the battery size required to handle 20% extra weight, designers might use robust optimization to calculate the maximum level of uncertainty allowable for an airplane capable of carrying a specific payload.

That our interviewees used GPkit primarily during conceptual design stages made the detailed accounting for uncertainty of robust optimization seem less necessary to them. In order to use robust optimization, they would have to create models with increased complexity in both concept and form, more difficult to interpret and to code. Some practitioners were additionally skeptical that doing so would significantly improve conceptual designs, as the uncertainties known at such an early stage felt more "made up" than other design parameters. While they found current uncertainty accounting practices to be more arbitrary, they felt that the specific uncertainty values they would choose in robust optimization might be just as arbitrary without the benefit of following industry standards. This formed the question for our human-subjects experiment: can robust optimization be useful (in comparison to current practices) even with guessed parametrizations of uncertainty?

4 HUMAN-SUBJECTS EXPERIMENT

This experiment was held to provide a direct comparison between methods of accounting for uncertainty with different computational models. We wanted in particular to see how additional uncertainty information mathematically encapsulated in models might shape designer's practices.

4.1 Methods

Forty-three graduate and undergraduate students in science and engineering were recruited to individually participate in a design challenge using a custom built graphical interface for a GPkit design model. Participants were prompted to choose parameters for an airplane design which led to designs with both as low a failure rate and as low a fuel consumption as possible. They were tasked with finding designs in three "reward regions" and to find designs on the final combined Pareto frontier; participants received greater compensation depending on their performance on these metrics. Each participant was given a ten minute tutorial, thirty minutes to complete the design challenge, and ten minutes to complete a short survey about their experience using the tool after the experiment. The code used for this experiment is available in an open source GitHub repository[1].

TABLE 2. PARTICIPANT DEMOGRAPHICS (SELF-REPORTED) Each participant was randomly assigned to an experimental condition upon arrival; no stratification was used to split participants.

Gender	*Female*	*Male*		
	21	22		
Education	*Undergrad*	*Masters*	*PhD*	
	23	9	11	
Department	*CS*	*Aero*	*Mech E*	*Other*
	13	14	12	4

4.1.1 Experimental Interface

The graphical interface shown in Figure 1 allowed users to directly modify a small set of parameters with sliders (A), then optimized a design based on those parameters and presented its fuel consumption (performance) and simulated failure rate. Participants kept track of the history of their designs with a plot of each design's fuel consumption and failure rate (B), a list of parameter combinations they'd tried that led to infeasible designs (C). The three reward regions were also shown on (B), providing a visual reminder of their goals. Additionally, participants saw the planform of their most recent airplane design (D). Sliders had discrete step values, but allowed arbitrary precision via typing. Fuel consumption was evaluated by solving the GPkit design model for the input slider values, while failure rate was determined by checking the model's feasibility across a set of one hundred randomized conditions; conditions were sampled from a multivariate truncated Gaussian probability distribution. A fixed set was used for all participants to enable comparability between the failure rates of various designs. This method of determining failure rates is similar to best-practices Monte Carlo simulations. The design model underlying this graphical interface was based on the "SimPleAC" GPkit model for passenger aircraft, [32] itself a condensed version of previous GPkit models for commercial aircraft [33, 34] that had been co-developed with the robust opti-

[1] https://github.com/convexengineering/robust_experiment

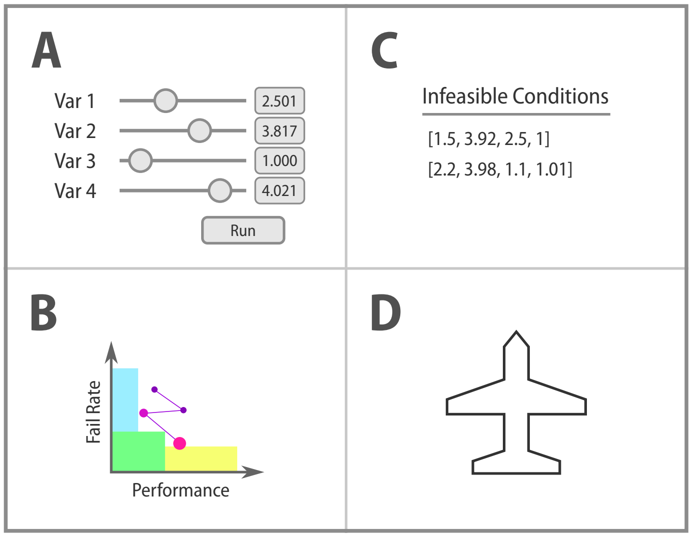

FIGURE 1. MOCK-UP OF EXPERIMENTAL UI
The three reward regions highlighted in the plot are designs with a fuel consumption below 1100 lbs (in blue), designs with failure rate below 10% (in yellow), and designs with both a fuel consumption below 1200 lbs and a failure rate below 30% (in green). The ordering of participant's designs was tracked through a line, with the most recent points appearing bright pink and older points appearing dark purple. A screenshot of the UI is in Appendix B.

mization library [3].

4.1.2 Experimental Conditions

Subjects were randomly partitioned into the four experimental conditions: two conditions similar to existing practices (Control and Margin), and two using robust optimization (Gamma Slider and Performance Slider). Participants using Control chose design parameters such as wing size; those using Margin chose safety factors, those using Gamma Slider chose the precise shape and scale of the uncertainty region they were optimizing for, while those using Performance Slider, chose the shape of that region and a desired performance, letting the optimizer maximize the scale of the uncertainty region. The uncertainty region was set to be elliptical, which represents a percentage of combined uncertainty being accounted for. Both Control and Margin represent current design practices: Control simulates common practices with non-optimizing design models, while Margin simulates common practices with GPkit models. Gamma Slider and Performance Slider represent the intended design practices Robust enables. We expected to see improvements to design space exploration coverage and quality with these uses of robust optimization.

More specifically, Control users directly manipulated four physical design parameters of the airplane (wing length, wing area, fuel volume available, and lift coefficient). Margin, Gamma Slider, and Performance Slider users directly manipulated parameters which accounted for uncertainty (margins or percentages of uncertainty on wing weight, fuel quality, takeoff speed, and range). Control users saw the fuel consumption of their designed airplane in the context it was optimized for, while users of the other design models saw performances which "priced in" uncertainty. Since the reward regions were identical across conditions, a larger fraction of possible designs Control users were able to find appeared in these regions. This kind of biased comparison is common in robust optimization practice. To compare performance across conditions during the analysis, designs made in non-Control conditions were "nominalized" by recalculating performance of each design in the nominal conditions Control designs had seen.

TABLE 3. SUMMARY STATISTICS
Given as intervals of a standard deviation around the mean. Significant differences (Pairwise t-test, Holm–Šidák correction, $p < 0.05$) indicated by asterisks. "Points" refers to feasible designs generated within thirty minutes. "% in R.R." refers to the percent of nominalized designs created in any of the regions participants were given financial incentive to be in. "Combined" refers to the average number of points found by each participant in each condition that were on the combined Pareto frontier across all conditions. ANOVA testing shows significance in all three statistics (Points: $p = 0.036$, % in R.R.: $p < 0.001$, Pareto $p = 0.006$).

	Points	% in R.R.	Combined
Control (N=10)	64 to 146	33% to 71%	0 to 0.4
Margin (11)	62 to 126	47% to 79%	0 to 1.7
Gamma S. (11)	103 to 163	*76% to 88%**	*0 to 4.2**
Perf. S. (11)	84 to 208	*77% to 91%**	*0 to 5.3**

4.2 Results

Prior to analyzing the quantitative data of the experiment, we assessed our overall impressions of each of the conditions from piloting and from informal post-experimental conversations with participants. Participants in the Control condition seemed to have the most direct understanding of how or why their parameter changes affected performance and failure rate, especially if they had some experience with airplane design. Participants in the Margin condition found their designs highly sensitive to even small parameter changes; it seemed easy to accidentally go to extremes with this tool. For both Performance Slider and Gamma Slider participants it seemed difficult to find designs far away from the Pareto frontier. Performance Slider participants could, by keeping the performance slider consistent, constrain their motion on the results plot to a single vertical line, allowing them to

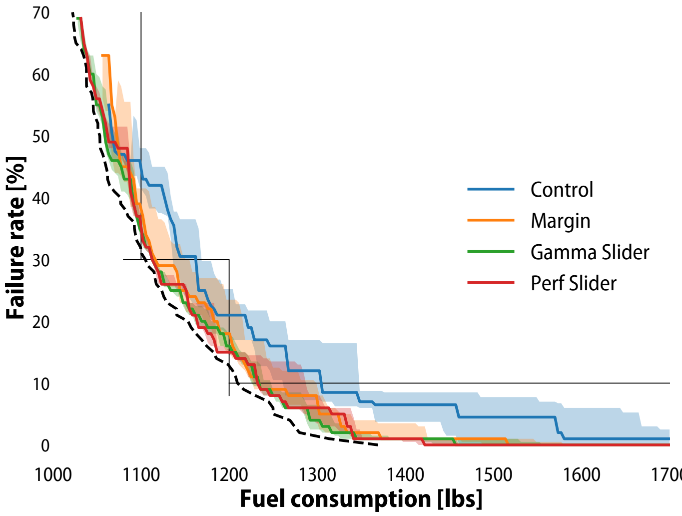

FIGURE 2. DISTRIBUTION OF FUEL WEIGHTS
Solid lines show median of participants' Pareto frontiers after nominalization. Shaded regions extend above it to the 75th percentile and below to the 25th. The black dashed line shows the combined final Pareto frontier, while solid black lines indicate reward regions.

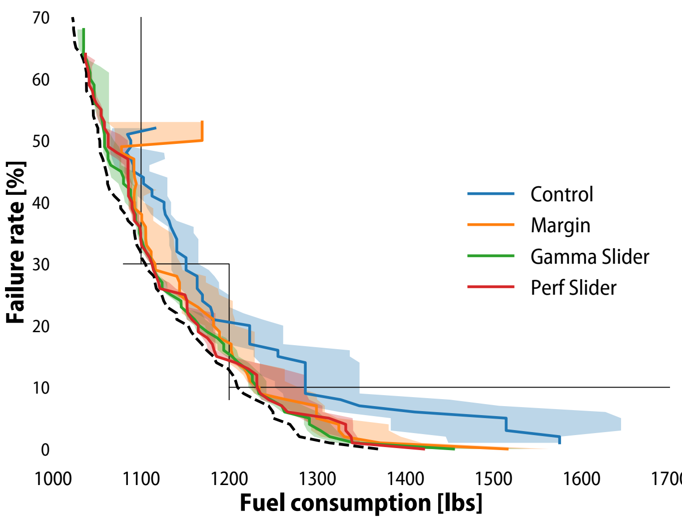

FIGURE 3. DISTRIBUTION OF FAILURE RATES
Solid lines show median of participants' Pareto frontiers after nominalization. Shaded regions extend to its right to the 75th percentile and to its left to the 25th. The black dashed line shows the combined final Pareto frontier, while solid black lines indicate reward regions.

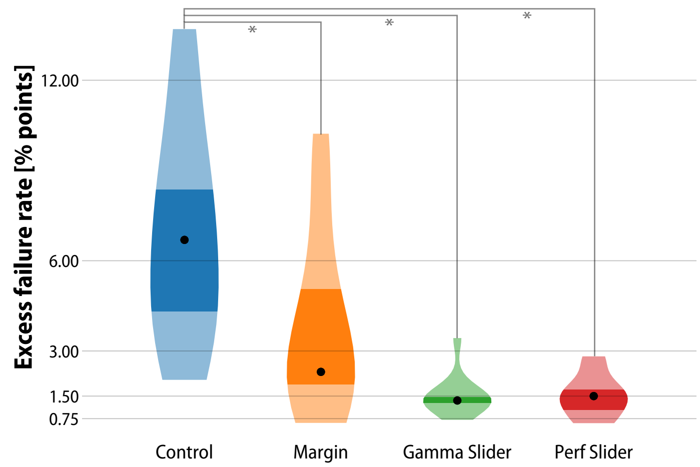

FIGURE 4. AVERAGE EXCESS FAILURE RATES
Significant differences (Pairwise t-test with Holm–Šidák correction, $p < 0.05$) indicated by an asterisk. Shaded region shows the distribution for each condition, darker between the 25th and 75th percentiles. Black dots show medians. ANOVA testing shows significance across conditions ($p < 0.001$).

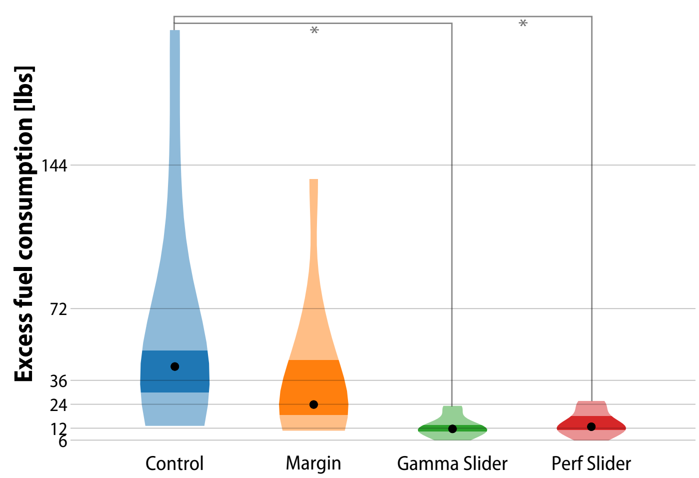

FIGURE 5. AVERAGE EXCESS FUEL CONSUMPTION
Significant differences (Pairwise t-test with Holm–Šidák correction, $p < 0.05$) indicated by an asterisk. Shaded region shows the distribution for each condition, darker between the 25th and 75th percentiles. Black dots show medians. ANOVA testing shows significance across conditions ($p = 0.007$).

separate dimensions inextricably linked for other users. Gamma Slider participants could, by keeping their standard deviations constant and only modifying the size of their uncertainty set, move along a single curve. While all conditions worked with four coupled variables, the addition of a uncoupled variable appears to have simplified the design task by reducing its dimensionality. Being able to act in only one "dimension" in these ways seemed to make the challenge less stressful for both Gamma Slider and Performance Slider participants.

To see if these impressions were validated by our data, we analyzed qualitative results from the post-experiment survey, which gave participants a set of statements and asked them to rate how much they agreed or disagreed with each on a six point Likert scale. Comparisons between Control and other conditions were biased by Control's easier access to the goal regions; given this, the fact that Control felt less stressed and frustrated than most other conditions is less surprising. Between other conditions, we saw differences in the amount participants felt like they "had a plan", were "in control", were "frustrated", or were "stressed". As expected, robust optimization conditions were less stressful and frustrating than Margin. However, Gamma Slider participants felt the least like they had a plan and were in control. This may indicate confusion about the "Gamma" parameter, which, as a robust-optimization specific term, lacked familiarity. Despite this, Gamma Slider participants had the highest quality solutions of all conditions. Even without feeling they understood what they were doing, Gamma Slider participants were able to find high quality designs.

The rest of this section quantitatively compares solutions across all four conditions. The design challenge incentivized participants not to find an optimal solution given a single goal, but rather to find a Pareto frontier of optimal solutions in terms of two goal parameters, performance and failure rate. To statistically analyze the influence conditions had on design outcomes, we compare the quantity of high quality points found in Table 3. The metrics of Pareto points and combined Pareto points serve as proxies for how much of the space was covered; the percent inside reward regions serves as a proxy for design quality. We see significant differences between robust optimization methods and standard methods in these metrics, providing evidence for the hypothesis that robust optimization encourages more exploration of optimal designs and increases the quality of each design explored.

The number of points metric is an indication of how much exploration participants were willing to do given specific tools; the large number of points in robust conditions indicates that exploration was faster and/or participants more willing to explore. The end times did not show significant differences, as there was no incentive to finish early. The tool itself did not take additional time to run in the Control or Margin conditions—in fact, it may have been slightly slower in the Robust conditions. The Control condition, where the reward regions were the easiest to achieve, provided less financial incentive to explore, which may have discouraged exploration. However, the Margin condition rated as slightly more stressful and frustrating due to its lack of predictability; participants may have been disincentivized to explore by stress or frustration, or may have required more time to determine the next point to test. A benefit of robust optimization may be either reduced stress and frustration or more intuitive changes in design quality, both leading to increased iteration.

We parametrize a design's quality with two dimensions: the improvement in failure rate that could have been achieved for that design's performance (vertical distance on the following plots), and the improvement in performance that could have been achieved for its failure rate (horizontal distance). In both cases, designs were compared to the final combined Pareto frontier achieved by other participants. Figures 2 and 3 show the distribution of these distances across participants' Pareto frontiers. Because we used the same reward regions across conditions, the difficult central region became therefore a focal point for some participants, as can be seen in the compression of their distribution at that point. With normalized performance, Control and the least-performant half of Margin participants are clearly separated from the combined frontier, while other participants are quite close.

To see the differences between the Pareto frontiers achieved by participants under condition, we summarize each individual frontier by its average vertical distance (Figure 4) and horizontal distance (Figure 5). We consider individual's frontiers all together instead of each of their points because such frontiers are the primary output of design model use, not a particular design point. That is, our simplified framework for the use of these models in a design process is 1) a condition is selected, 2) a Pareto frontier created, and 3) a condition is chosen from that Pareto frontier based upon the whole frontier.

Figure 4 shows the distributions of excess failure rates (average vertical distance) across the frontiers made with each condition. There is a clear distinction between Control and Margin, and between both of them and the two robust conditions. Figure 5 shows the distribution of excess fuel consumption (average vertical distance) across conditions. The frontiers of median users of the robust models perform better by this metric than the best users of Margin and Control, and every user of robust models performs better than three quarters of Control users.

5 DISCUSSION

These results are evidence that robust optimization can increase design quality. Returning to our fundamental research questions, what do they imply about the effects of conceptualizations and formulations of uncertainty, and what current design practices might robust optimization alter or automate?

5.1 RQ1: Conceptualization of Uncertainty

From practitioner interviews we found that uncertainty conceptualization in the early stages of airplane design is minimal, partly because uncertainty is considered fruitless to estimate when the overall design is rapidly changing. However, we found two types of uncertainty were being mixed together: 1) uncertainty related to changes that were part of the design process, and 2) uncertainty related to the range of possibilities the final design might face. The conceptual merging of these meant that designers who did not think they could account for the first type, also thought they could not account for the second. For robust optimization to be used in conceptual design, it must make clear it is formulated for the second type.

Given that designers at this stage do not often conceptualize this second type of uncertainty, how might they adopt robust optimization? Experimental participants in the robust Performance Slider condition felt most like they "had a plan"; Gamma Slider participants felt least like they had a plan. This implies that, for non-expert users, the terminology of robust optimization (present in Gamma Slider as the "Gamma" factor, but absent in Performance Slider) may be a barrier to entry. However, the concept of optimizing for uncertainty, present in both conditions, did not seem to hinder understanding (using "felt like they had a plan" as a proxy). For GPkit users trying robust optimization, we would expect the transition to be eased by parallels between the conceptualization of uncertainty in robust optimization and uncertainty questions already asked later in the design process. The Performance Slider condition is analogous to finding the most robust design possible for a certain performance; the Gamma Slider condition is analogous to finding the most performant design possible for a specific uncertainty set. The additional complexity of design models in practice and the lack of GUI-based abstraction may limit the generality of these results.

5.2 RQ2: Formulation of Uncertainty

The current process of GPkit model creation does not encourage a rigorous formulation of uncertainty. Practitioners discussed multiplying uncertain fixed variables with industry-standard safety factors, but this method seemed more of a default practice rather than one engaged with a conceptualization of uncertainty.

In our experiment, the Control condition had no formulation of uncertainty, the Margin condition encapsulated uncertainty in safety factors, and the robust optimization conditions encapsulated uncertainty in relative standard deviations. Results showed participants in Control and Margin were far worse at finding Pareto optimal points than participants in robust optimization conditions: 75% of robust optimization frontiers were better than the median frontier of the other conditions. Additionally, formulating uncertainty as a directly controllable variable seems to have reduced the quantity of suboptimal designs explored.

In this simplified design challenge, the model's formulation was abstracted away from the participants. In practice, users of GPkit would need to understand robust optimization well enough to create these models on their own. While Robust was designed to only require a small amount of additional code, the mathematical increase in understanding needed to create such syntax was not accounted for within this study. It remains to be investigated as a possible obstacle to usage of robust optimization in GPkit.

5.3 RQ3: Automated Design Processes

Our experiment was designed to represent both designers' present design exploration processes and the potential processes of robust optimization. Our failure rate simulation was meant to mimic a designer testing their design, either through Monte Carlo simulation, more complex computational modeling, or prototype creation. In this study, this failure rate simulation formed the "ground truth" of the participants involved; in practice, the ground truth could not be so easily discovered at this stage. A simulation similar to ours would serve as an early check in the design process, rather than the final one.

Current design processes were emulated by the Control and Margin conditions. Control emulated the process of manually setting design parameters without use of optimization, as is common in conceptual aerospace design. Our results find that, while it is possible to find high quality solutions this way, it is difficult to do so consistently. Our Margins participants emulated the process of specifying safety factors within an optimization framework such as GPkit. Margins are not so flexibly set in practice. Instead, they are generally fixed at an industry-standard value. Similarly, simulations to check failure rates are more generally performed after a solution has been decided upon, not during a single designer's rapid iteration through designs. Both the Margin and Control conditions of our experiment put current practices on a much faster timescale; caution should be taken equating these results with current design practices. The optimization involved in Margin, as well as the ability to control uncertainty parameters, led to higher quality designs than those of Control participants, though Margin participants were still able to find poor quality designs far away from the Pareto frontier.

Judging just by what participants saw on their screen, the Control case had an easier time reaching the reward regions. However, this is due to the method in which uncertainty is incorporated into the mathematical model—since the uncertain variables are directly modified to be in their worst case of the uncertainty accounted for, the performance given by the model is the performance under worst case conditions. We presented this performance to participants to better simulate how designers would view each tool. To be able to compare the underlying data however, we needed to "nominalize" the data, which meant rerunning the model with optimized fixed design parameters with uncertainty parameters set to the nominal values used by the Control condition. This workflow on the experimenter's part implies the need for an automated functionality to compare designs optimized for various conditions; practitioners also noted the need to easily test performance on "off-design" cases.

The Gamma Slider and Performance Slider conditions mimic two ways designers could use robust optimization to explore the design space, and the consistent quality of their Pareto frontiers implies that the methods can produce a high likelihood of Pareto optimality without requiring much skill. Given the mathematical formulation of robust optimization, this is no surprise. A random sample of conditions is an approximation of the bounds robust optimization is designed to optimize for; the failure rate returned by the random sample is a less accurate representation of how much uncertainty is accounted for than the robust optimization's own parameter bounds. This turns the experiment into a game of finding uncertainty parameters that overfit the controlled set of one hundred random samples. A designer mimicking this process in practice would set the bounds of both the Monte Carlo simulation and the uncertainty parameters of robust optimization; however, a probabilistic simulation analysis does not make sense if the designer can choose the space of uncertainty optimized for. Robust optimization automates away the mathematical necessity of performing Monte Carlo simulations over direct design parameters.In practice, we would expect Monte Carlo simulations to still be used to provide additional legitimacy to designs for stakeholders with less familiarity with robust optimization practices, and for uncertain parameters not representable within a convex model.

Robust optimization's most apparent advantage becomes clearer later in the design process—the expressivity it provides designers to build models that are detailed mirrors of their project-specific conceptions of uncertainty [35]. However, this potential benefit would require a change in how GPkit is used; while some designers wanted to continuously update GPkit models as their designs proceeded past the conceptual stage, they felt little ability or incentive to do so, as their coworkers usually trusted more complex "high-fidelity" to be more legitimate.

Trust in GPkit models of various designs does need to be built; not many designers would be willing to use the values determined as optimal directly from a GPkit solve without first validating the model in other software. However, late-stage GPkit models have been able to accurately predict the performance of an airplane prototype, such as with the Jungle Hawk Owl [36,37], whose designers built a plane fully modelled in GPkit, and found their built performance remarkably close to model estimates. However, to encourage adoption of robust optimization in GPkit, improvements in design quality must be evident even at early conceptual stages. This study provides evidence that robust optimization can have a dramatic effect, even with a simple conceptual model.

6 CONCLUSION

This study provides evidence for the importance of accounting for uncertainty early in the design process. A lack of uncertainty formulation within a design model can require external, imperfect metrics of uncertainty testing, such as Monte Carlo simulations, and the iteration modeling process is thus less likely to produce high quality designs. Simple uncertainty formulation within a design model, such as multiplying a variable by a safety factor, can create overly conservative designs or make worthwhile designs appear infeasible. However, most designers do not know alternative methods of accounting for uncertainty, or consider those methods to be impractical for conceptual design.

Robust optimization provides stronger protections against uncertainty than safety factors, making it difficult for even inexperienced users to create non-robust designs. This is seen through the high quality of almost all our experimental participants' final designs relative to the combined Pareto frontier. We also provide two conceptualizations of uncertainty GPkit users could use robust optimization to represent. The first, represented by Performance Slider, is optimizing for the largest scaled uncertainty, creating an airplane that is as robust as possible for a particular performance. The second, represented by Gamma Slider, is optimizing for performance, creating an airplane that maintains a particular level of robustness while spending little on fuel. GPkit users who already consider uncertainty via Monte Carlo simulations of their designs will find robust optimization essentially automates the function of Monte Carlo simulation within it, reducing the necessity of running additional simulations on designs.

The human-subjects experiment was a game for novices, and so does not allow us to draw conclusions about how designers in practice might behave. However, even though robust optimization uncertainty parameters were difficult to understand conceptually, this barrier did not prevent novice participants from finding high quality solutions. The experiment also provides questions for future field studies: Do explicit formulations of uncertainty enable better conversations about it during conceptual design? How do multiple stakeholders interact with these tools and solutions to reach an agreement? Do the benefits found in this study extend to more complex solutions? How difficult is it for designers to transition from formulating uncertainty as safety factors to skillfully using robust optimization? Answering these questions will allow us to understand the potential of robust optimization as a method for accounting for uncertainty.

ACKNOWLEDGMENTS

This study could not have been conducted without the intellectual and emotional support of these people, who helped us overcome the variety of challenges of this project.

First, the authors would like to thank all of the pilot testers, participants, and interviewees of the study. Your excitement and curiosity made our jobs a lot easier and more enjoyable.

We would also like to thank Berk Ozturk and David S. Anderson for helping with the conceptual development of this study and for using their knowledge of optimization to help us work through some of the more confusing mathematical quirks of Robust and GPkit, as well as Ali Saab, without whom Robust GPs would be but a daydream.

Finally, we would like to thank Jana Saadi, Shiroq Al-Megren, and Sujithra Raviselvam for providing helpful perspective and making sure lab was always a fun place to be.

This material is based upon work supported by the National Science Foundation under Grant No. 1854833 and the MIT-SUTD International Design Center.

REFERENCES

[1] Burnell, E., Damen, N. B., and Hoburg, W., 2020. "GPkit: A Human-Centered Approach to Convex Optimization in Engineering Design". In Conference on Human Factors in Computing Systems (CHI), Association for Computing Machinery, p. 12.

[2] Bertsimas, D., Brown, D. B., and Caramanis, C., 2011. "Theory and applications of robust optimization". *SIAM Review,* ***53***(3), pp. 464–501.

[3] Öztürk, B., and Saab, A., 2019. "Optimal Aircraft Design Decisions under Uncertainty via Robust Signomial Programming". In AIAA Aviation 2019 Forum, AIAA, p. 29.

[4] Saab, A., Burnell, E., and Hoburg, W. W., 2018. "Robust Designs via Geometric Programming". *arXiv,* ***1808.07192***, p. 23.

[5] Pahl, G., and Beitz, W., 2013. *Engineering design: a systematic approach*. Springer Science & Business Media, New York.

[6] Ulrich, K. T., Eppinger, S. D., and Yang, M. C., 2020. *Product Design and Development*. McGraw-Hill Education, New York.

[7] Leonardi, P. M., 2011. "When flexible routines meet flexible technologies: Affordance, constraint, and the imbrication of human and material agencies". *MIS Quarterly,* ***35***(1), pp. 147–167.

[8] Sutherland, I. E., 1964. "Sketchpad: a man-machine graphical communication system". *Simulation,* ***2***(5), p. 18.

[9] Parkin, K. L., Sercel, J. C., Liu, M. J., and Thunnissen, D. P., 2003. "Icemaker$^{\mathrm{TM}}$: an excel-based environment for collaborative design". In IEEE Aerospace Conference, IEEE, p. 11.

[10] Nardi, B. A., and Miller, J. R., 1991. "Twinkling lights and nested loops: distributed problem solving and spreadsheet development". *International Journal of Man-Machine Studies,* ***34***(2), pp. 161–184.

[11] Mark, G., 2002. "Extreme collaboration". *Communications of the ACM,* ***45***(6), pp. 89–93.

[12] Robertson, B., and Radcliffe, D., 2009. "Impact of CAD tools on creative problem solving in engineering design". *Computer-Aided Design,* ***41***(3), pp. 136–146.

[13] Buenza, D., and Stark, D., 2004. "Tools of the trade: the socio-technology of arbitrage in a Wall Street trading room". *Industrial and Corporate Change,* ***13***(2), pp. 369–400.

[14] Greenwood, D. J., and Levin, M., 2006. *Introduction to Action Research: Social Research for Social Change*. SAGE Publications, Thousand Oaks.

[15] Fixson, S. K., and Marion, T. J., 2012. "Back-loading: A potential side effect of employing digital design tools in new product development". *Journal of Product Innovation Management,* ***29***, pp. 140–156.

[16] Mahan, T., Meisel, N., McComb, C., and Menold, J., 2018. "Pulling at the Digital Thread: Exploring the Tolerance Stack Up Between Automatic Procedures and Expert Strategies in Scan to Print Processes". *Journal of Mechanical Design,* ***141***(2), 12. 021701.

[17] Hirschi, N., and Frey, D., 2002. "Cognition and complexity: an experiment on the effect of coupling in parameter design". *Research in Engineering Design,* ***13***(3), pp. 123–131.

[18] Flager, F., Gerber, D. J., and Kallman, B., 2014. "Measuring the impact of scale and coupling on solution quality for building design problems". *Design Studies,* ***35***(2), pp. 180–199.

[19] McComb, C., Cagan, J., and Kotovsky, K., 2015. "Rolling with the punches: An examination of team performance in a design task subject to drastic changes". *Design Studies,* ***36***, pp. 99–121.

[20] Austin-Breneman, J., Honda, T., and Yang, M. C., 2012. "A study of student design team behaviors in complex system design". *Journal of Mechanical Design,* ***134***(12), p. 4.

[21] Austin-Breneman, J., Yu, B. Y., and Yang, M. C., 2016. "Biased information passing between subsystems over time in complex system design". *Journal of Mechanical Design,* ***138***(1), p. 9.

[22] Yu, B. Y., 2015. "Human-centered approaches to system level design with applications to desalination". PhD Thesis, Massachusetts Institute of Technology.

[23] Neeley, W. L., Lim, K., Zhu, A., and Yang, M. C., 2013. "Building fast to think faster: exploiting rapid prototyping to accelerate ideation during early stage design". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE), American Society of Mechanical Engineers.

[24] Brady, J. T., 1986. "A theory of productivity in the creative process". *IEEE Computer Graphics and Applications,* ***6***(5), pp. 25–34.

[25] Scott, S. D., Lesh, N., and Klau, G. W., 2002. "Investigating human-computer optimization". In Conference on Human Factors in Computing Systems (CHI), pp. 155–162.

[26] Mueller, C., and Ochsendorf, J., 2013. "An integrated computational approach for creative conceptual structural design". In IASS Annual Symposia, Vol. 2013, International Association for Shell and Spatial Structures, pp. 1–6.

[27] Barron, K., Simpson, T. W., Rothrock, L., Frecker, M., Barton, R. R., and Ligetti, C., 2004. "Graphical user interfaces for engineering design: impact of response delay and training on user performance". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE), American Society of Mechanical Engineers, pp. 11–20.

[28] Egan, P., Cagan, J., Schunn, C., and LeDuc, P., 2015. "Synergistic human-agent methods for deriving effective search strategies: the case of nanoscale design". *Research in En-*

gineering Design, ***26***(2), pp. 145–169.
[29] Boyd, S., Kim, S.-J., Vandenberghe, L., and Hassibi, A., 2007. “A tutorial on geometric programming”. *Optimization and Engineering,* ***8***(1), pp. 67–127.
[30] MOSEK ApS, 2014. The MOSEK optimization APIs for C and Python.
[31] Andersen, M. S., Dahl, J., and Vandenberghe, L., 2013. CVXOPT: A Python package for convex optimization.
[32] Öztürk, B., 2018. “Conceptual engineering design and optimization methodologies using geometric programming”. Master’s thesis, Massachusetts Institute of Technology.
[33] Kirschen, P. G., Burnell, E., and Hoburg, W., 2016. “Signomial programming models for aircraft design”. In 54th AIAA Aerospace Sciences Meeting, AIAA, p. 26.
[34] York, M. A., Öztürk, B., Burnell, E., and Hoburg, W. W., 2018. “Efficient Aircraft Multidisciplinary Design Optimization and Sensitivity Analysis via Signomial Programming”. *AIAA Journal,* ***56***(11), pp. 4546–4561.
[35] Burnell, E., Pillai, P. P., and Yang, M. C., 2020. “Maps, Mirrors, and Participants: Design Lenses for Sociomateriality in Engineering Organizations”. In Conference on Designing Interactive Systems (DIS), Association for Computing Machinery, p. 9.
[36] Burton, M. J., and Hoburg, W. W., 2017. “Solar and Gas Powered Long-Endurance Unmanned Aircraft Sizing via Geometric Programming”. In Multidisciplinary Analysis and Optimization Conference, AIAA/ISSMO, p. 14.
[37] Burton, M., Hansman, R. J., Tao, T., and Hoburg, W., 2019. “Flight Test Report of the Jungle Hawk Owl Long-Endurance UAV”. *ICAT Report,* ***2018-09***.

Appendix A: Questionnaire for Interviews

Questions were grouped into three broad categories:

(A) Background/General
(B) Integration/Communication
(C) Robustness

Questions were given approximately in this order, allowing for flexibility given the natural flow of conversation.

1. Tell me about the projects you are working on and your role within them. **(A)**
2. How and why did you start using GPkit? **(A)**
3. Think about a project you did that could have used GPkit, but didn’t.
   (a) Why did the project not use GPkit? **(A)**
   (b) How did you integrate and optimize your systems? What tools did you use to integrate and optimize your systems? **(B)**
   (c) How long did the design process take? How many early stage iterations (i.e. early simulations) did you go through? How many late stage iterations (i.e. more detailed simulations, built objects) did you go through? **(C)**
   (d) How closely did early simulations match the final object? **(C)**
   (e) How many people were involved? How were they organized? **(B)**
   (f) How did you evaluate the quality of your design during the process? After it was complete? **(C)**
4. Think about the last project you did with GPkit.
   (a) What stages of the project did you use GPkit during? **(B)**
   (b) How did you use GPkit to integrate and optimize your systems? What processes did GPkit replace, and which ones did it not replace? **(B)**
   (c) What tools did you use in addition to/before/after GPkit? **(B)**
   (d) How long did the design process take? How many early stage iterations (i.e. early simulations) did you go through? How many late stage iterations (i.e. more detailed simulations, built objects) did you go through? **(C)**
   (e) How closely did early simulations match the final object? **(C)**
   (f) How many people were involved? How were they organized? **(B)**
   (g) How did you evaluate the quality of your design during the process? After it was complete? **(C)**
5. Of the differences in the two projects we mentioned, which ones were related to GPkit? **(A)**
6. If you haven’t used GPkit in major projects, why? **(A)**
7. What do you view as benefits of GPKit? **(A)**
8. What do you find to be lacking in GPkit? What features would you like to be added? **(A)**
9. What qualities of a project do you find make it better suited for GPkit? **(A)**
10. How do you moderate uncertainty? (i.e. do you prioritize accuracy in measurements of certain components versus others?) **(C)**
11. How do you encode uncertainty information into GPkit? **(C)**
12. How does your initially designed model translate into the final built structure? What things change? How often are you re-solving your model/modifying the design? **(C)**

Appendix B: Experimental UI

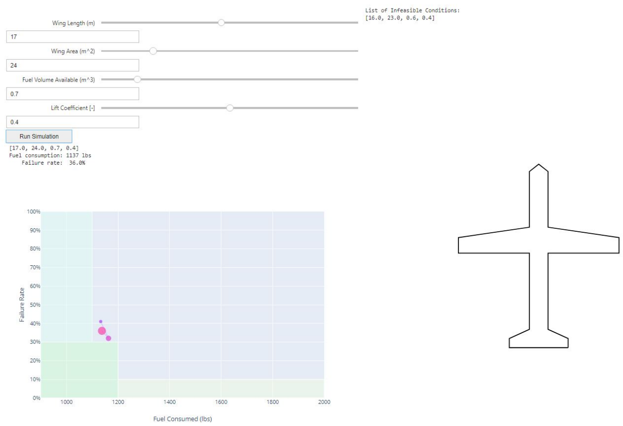

FIGURE 6. Experimental UI
Screenshot of interface seen by participants of the human-subjects experiment.