AN INVESTIGATION OF PRESCRIBED RISK MANAGEMENT PRACTICES IN ENGINEERING DESIGN

A Thesis
Presented to
The Academic Faculty

By

Benjamin David Lee

In Partial Fulfillment
of the Requirements for the Degree of
Master of Science in Mechanical Engineering

Georgia Institute of Technology

May 2010

AN INVESTIGATION OF PRESCRIBED RISK MANAGEMENT PRACTICES IN ENGINEERING DESIGN

Approved by:

Dr. Chris Paredis, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Leon F. McGinnis
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Baabak Ashuri
College of Architecture
Georgia Institute of Technology

Date Approved: April 1, 2010

ACKNOWLEDGEMENTS

I would like to thank Nicole Miller, for motivating me to complete this thesis sooner rather than later. Thank you for tirelessly listening to my rants and always being there with your love, support, and encouragement.

I would also like to thank my advisor, Dr. Chris Paredis, for his guidance and inspiration. Thank you for working so hard to ensure that we could always see the larger picture, as well as the small details.

I would also like to thank my friends and colleagues for their support, knowledge, and conversations which served to aid in the completion of my Master's Thesis: Aditya Shah, Alek Kerzhner, Roxanne Moore, Stephanie Thompson, Rich Malak, Kevin Davies, Patrick Chang, and Mukul Singhee. I would also be remiss not to thank the rest of the Systems Realization Laboratory for their inspirational and informative discussions, support, and friendship.

I would like to thank my reading committee, Dr. Baabak Ashuri and Dr. Leon McGinnis, for taking the time to read and offer their insights on this work.

I would like to thank the NSF Center for Compact and Efficient Fluid Power and the National Science Foundation- Grant No. CMMI-0522116 for their generous support in funding this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

In this thesis, a decision model for examining prescribed risk management practices in engineering design is presented. The decision model explicitly considers the effects that design decisions under uncertainty have on the overall utility of the design process. These effects are important to consider because, according to Utility Theory, the designer should make decisions such that the expected utility is maximized. However, a significant portion of the literature neglects the costs of the design process, and focuses only on the quality of the design artifact, or at best includes its manufacture when determining the utility of an alternative. When designers neglect the costs of the design process, they cannot make tradeoffs between the costs of the design process and the quality of the artifact. As compared to previous work in this area, the decision model presented includes the effects of temporally degrading product utility on design decisions. The decision model is used to investigate the impacts of degrading product utilities in products that launch later as a result of the duration of design actions performed. In this thesis, the decision model is leveraged to investigate two key trends in engineering design resulting from increasing temporally-based costs. To support the conclusions in this thesis, quantitative evaluations of the decision model are investigated for two case studies. The conclusions are additionally supported through evaluations of the decision model in boundary plots that visualize prescribed behavior for designers over varying model parameters.

CHAPTER 1

INTRODUCTION

1.1 Problem statement

Engineering design is a complex process comprised of several tasks which are usually performed iteratively. Figure 1 shows a simple model of a design process in which a designer, given a problem description, iteratively performs the activities of *Ideation, Analysis, Evaluation*, and ultimately *Selection* of the final design.

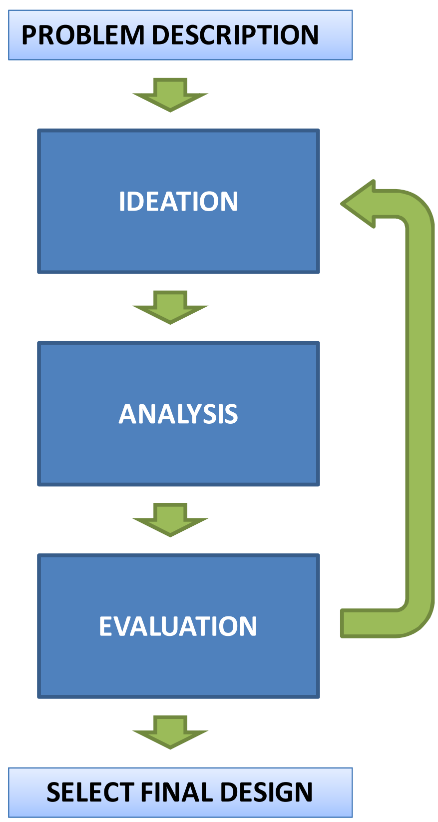


Figure 1. A Simple Model of an Iterative Design Process

Of particular importance in this process are the activities of *Analysis* and *Evaluation*, as these tasks are responsible for steering the designer towards what is deemed as preferable. In the context of this thesis, *Analysis* and *Evaluation* are viewed as separate processes because of the desired separation between beliefs and preferences in engineering design. *Analysis* is a form of beliefs, as an analysis is a design action that the designer believes will yield additional information about a particular design alternative: Finite-Element Modeling, Prototyping, Marketing Analysis, Back of the Envelope Calculations, etc. *Evaluation* is a form of preferences, as an evaluation is an explicit comparison of a design alternative to the designer's preferences used to determine a measure of effectiveness for the design alternative. A design alternative should always be *Evaluated* after it is *Analyzed*, otherwise the designer does not gain any new information as its suitability.

Previous work by Thompson and Paredis [1] investigated trends in the selection of these design activities using a prescriptive decision model and a process utility-driven measure of effectiveness. However, this work neglects an important feature of analyses; in addition to the monetary resources required, they also require time resources. Since the utility of many products, especially novel products, tends to be time-sensitive, the expected utility of a particular analysis may be degraded as well if its duration is accounted for.

1.2 Motivating Question and Hypotheses

The time-sensitive nature of product utility introduces the following question,

How does the selection of design process activities depend on temporal degradation of product utility?

In this thesis, two hypotheses are posed as responses to the motivating question.

H1: When considering the temporal degradation of product utility, the maximization of expected utility leads to the parallelization of design tasks.

This first hypothesis is based on the belief that as time becomes increasingly valuable, designers will need to gather information about design alternatives more rapidly to make well-informed decisions.

H2: When considering the temporal degradation of product utility, the maximization of expected utility leads to risk acceptance rather than risk mitigation

This second hypothesis is based on the belief that as the temporal costs of analyses increase, they will at some point become so expensive that their costs will outweigh their expected benefits. This should be evidenced by reduced artifact testing and increased levels of uncertainty at design selection as temporal costs increase.

To investigate these hypotheses, the Temporal Analysis Decision Model (TADM) is presented and examined in this thesis. A decision model, typically visualized as a decision tree (see Figure 2), is a tool used to analyze the effects of decisions or series of decisions. Decision models can be used to determine the optimal series of decisions according to some measure of effectiveness, which makes them particularly useful for investigating scenarios where behavior is prescriptively defined by Utility Theory.

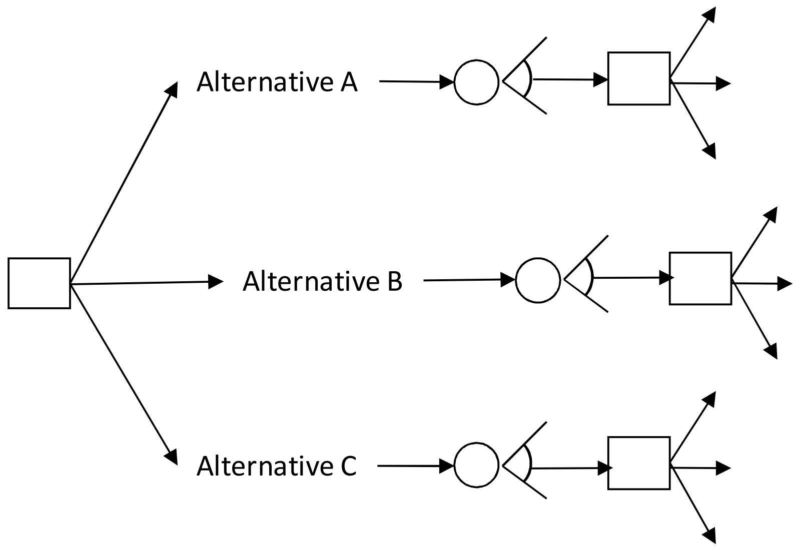


Figure 2. An Example of a Decision Model, Visualized as a Decision Tree

The Temporal Analysis Decision Model contains the design actions available to a designer (Analyze, Select) when faced with a decision about which design alternative to produce. Through investigations of how the prescriptively defined behavior of a designer changes as temporal costs increase, this thesis will leverage the decision model to support the hypotheses.

1.3 Thesis Organization

The remainder of this thesis is organized as follows: In the next chapter, engineering design is defended as an appropriate domain for the application of Utility Theory principles. This includes an investigation of the foundation of Utility Theory and an examination of methods for design under uncertainty. In Chapter 3, the relevant literature investigating the temporal effects on product utility is investigated. Then, the Temporal Analysis Decision Model is presented as a tool for investigating the hypotheses. Simplifying assumptions for the execution are introduced and examined. In Chapter 4, the TADM is applied to two case studies: an OEM parts supplier faced with a deadline, and a consumer electronics company in a competitive marketplace. Additionally, boundary plots are used to investigate variations in model parameters, and further support the hypotheses. In Chapter 5, the contributions of the thesis are summarized, and opportunities for future work are discussed.

CHAPTER 2

ENGINEERING DESIGN AS A SUITABLE DOMAIN FOR THE APPLICATION OF UTILITY THEORY

2.1 The Role of Decision Theory in Design

Many different frameworks have been proposed to describe the process of design [2-7]. These frameworks generally attempt to prescribe a particular method that designers should use to guide themselves through the design process. Decision-Based Design (DBD) is one such framework and is based on the viewpoint that design can be decomposed into a series of decisions [8]. DBD is based upon the mathematical foundation of decision theory provided by axiomatic Utility Theory [9]. DBD uses this basis to analyze series of decisions, and prescribes that the Decision Maker (DM) should perform the action that maximizes his or her expected utility. In the context of design, this means that the designer should perform the design actions such that the net utility of the design process is maximized. It should be noted that while product utility may have the greatest impact on net utility, the resources consumed during the design process may have significant impact as well. Thus, when considering the net utility, the designer should consider the consumption of these resources.

2.2 The Foundation of Utility Theory

Design decisions are often made under significant uncertainty, especially early in the design process. This uncertainty can come from various sources: how a product will sell in the marketplace, the cost of manufacturing, performance levels of the product in different scenarios, or many others. A result of the uncertainty is that several different outcomes for any given decision may be possible, which gives rise to risk in the design process. Risk is a concept that reflects the possible variation in a measure of effectiveness due to uncertainty, and is defined as the product of the probability of occurrence and consequences of the outcomes.[1]

An axiomatic theory for making design decisions under uncertainty is provided by Utility Theory [2], which states that the Decision Maker (DM) should select the alternative with the largest expected utility. Preferences under uncertainty can be expressed in terms of utilities, the properties of which are outlined in Utility Theory. The axiomatic foundation of Utility Theory was first developed by von Neumann and Morgenstern [9] (vN-M), and others have developed slightly differing sets of axioms that reach similar results [11-14]. The original axioms as set out by vN-M are reviewed in Figure 3.

NOTE: (u, v, w) are outcomes. (α, β) are probabilities. $u \succ v$ indicates that outcome u is preferred to outcome v. $u \sim v$ indicates that outcome u and v are equally preferred.

[1] It is important to note that two conflicting definitions of risk exist within the design community. In the classical definition, both negative and positive effects of variability are considered [2], whereas the second definition only considers the negative effects of variability [10] with the positive effects being credited as a separate windfall. In this thesis, the second risk definition is used, as it is the one more commonly used in practice.

1. *Complete Ordering*

For any (u, v) either $u \succ v$ OR $u \prec v$ OR $u \sim v$

2. *Transitivity*

For any (u, v, w) if $u \succ v$ AND $v \succ w$ THEN $u \succ w$

3. *Continuity*

For any (u, v, w) such that $u \succ w \succ v$, then for some

$\alpha, (0 < \alpha < 1), w \sim \alpha u + (1 - \alpha)v$

4. *Convexity*

For any (u, v) such that $u \succ v$, then for any α,

$(0 < \alpha < 1), u \succ \alpha u + (1 - \alpha)v$

5. *Combining*

For any (u, v) , $(0 < \alpha\beta < 1)$ and $\gamma = \alpha\beta$,

$\alpha(\beta u + (1 - \beta)v) + (1 - \alpha)v \sim \gamma u + (1 - \gamma)v$

Figure 3. Axioms of Utility Theory [9]

The first axiom states that the DM has preferences over any possible outcome, and that the DM is capable of expressing that preference. The second axiom states that preferences should be consistent and transitive. The remaining axioms concern the consideration of vN-M lotteries (see Figure 4). In a vN-M lottery, the DM has the option to enter into a lottery with uncertain outcomes $\mathrm{A}_1, \ldots, \mathrm{A}_\mathrm{n}$ ranked from most to least desirable, each with a corresponding probability of occurrence $\mathrm{p}_1, \ldots, \mathrm{p}_\mathrm{n}$. The third axiom states that preferences should be continuous over a region: any lottery with two outcomes as possibilities can be reduced to an equivalent certain outcome. The fourth axiom states that preferences should be convex: if something is preferable, an increased chance of receiving it should always be preferred. The fifth axiom states that compound lotteries, or lotteries with a lottery as an outcome, can be reduced to a single lottery.

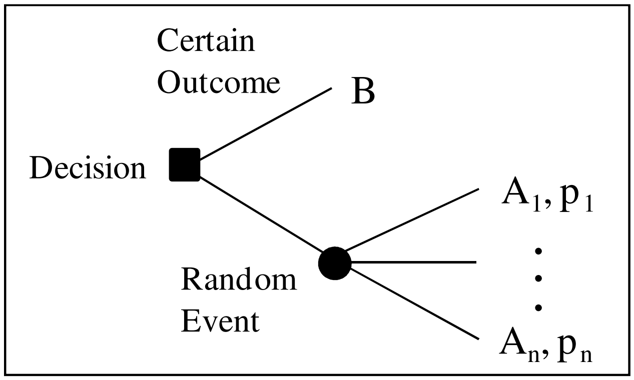


Figure 4. An example of A vN-M Lottery

The axiomatic foundation imposes simple limitations on the definition of utilities and establishes the rationality of the DM, protecting him or her from a sure loss. However, the axioms do not impose any preference models on the DM. Rather, it is recognized that decision making is a subjective process and the foundation allows for any set of preferences to be modeled so long as they are rational; i.e. they cannot account for a DM changing his or her mind on a whim.

2.3 Development of Utility Functions

As described in the previous section, Utility Theory is constructed from the consideration of vN-M lotteries. In this section, two formulations that organize a DM's preferences into a mathematical function are introduced. These utility functions make the comparison of multiple alternatives a simpler, more explicit process.

2.3.1 Single Attribute Utility Functions

The simplest formulation for a utility function is the single-attribute case. As shown in Eqns. (1-2), the utility is calculated by collecting all important parameters into a single all-encompassing attribute, and then determining the utility over that attribute. In Eqns. (1-2), X is the single attribute over which preference is elicited, X_i are the various parameters which define X through the transformation f. For example, if a new engine was being designed, X could be the projected profitability of an engine with horsepower=X_1, weight=X_2, cost=X_3, etc.

$$U = Pref(X) \qquad \textit{Eqn. (1)}$$

$$X = \boldsymbol{f}(X_1, X_2, \dots, X_n) \qquad \textit{Eqn. (2)}$$

Hazelrigg [2] advocated that designers should adopt an enterprise context, and use the single attribute formulation with profit being the primary driver of utility. He argued that "the goal of design is to make money, and more is better". At large, this formulation appears to provide a meaningful measure of effectiveness.

However, there are some scenarios where the profitability of an alternative may not be a proper measure of effectiveness. For example, scientific research is driven by the desire to create new knowledge, not revenue. For scenarios such as this where profit is not a sufficiently important driver to warrant sole consideration, a utility function considering multiple attributes may be appropriate.

2.3.2 Multiple Attribute Utility Functions

As opposed to the single attribute formulation, Multi-Attribute Utility Theory (MAUT) as developed by Keeney and Raiffa [15] encourages DMs to elicit preferences over several attributes, and then to combine those utilities using a multi-attribute utility function. There are some requirements which must be met to use the MAUT formulation; the attributes should be utility independent of each other, meaning that preference for vN-M lotteries over one attribute do not depend on the value of other attributes. For further discussion about how to determine mutual utility independence see [16]. Once the individual utility functions over each attribute have been elicited (Eqn. (3)), they are combined into a single utility function through a combination *g,* (Eqn. (4)). Based on the nature of the attributes under consideration, the function combining the attributes may take on many different forms: Multiplicative, Multi-linear, Additive, or others. As an example of a scenario where multiple attributes are important, consider the case where the living quarters for a manned base station for Mars is being designed: X_1 could be the volume of the living space, with X_2= cost, X_3= service life, etc.

$$U_{X_i} = pref(X_i) \qquad \textit{Eqn. (3)}$$

$$U(X_1, X_2, \dots, X_N) = g\left(U_{X_1}, U_{X_2}, \dots, U_{X_N}\right) \qquad \textit{Eqn. (4)}$$

2.4 Alternative Methods for Designing under Uncertainty

A major criticism of Utility Theory is that it is too complicated and arduous to elicit a designer's preferences and apply them in real engineering scenarios [17]. As a result, designers may instead utilize one of several methods for design under uncertainty such as Robust Design (RD), Reliability Based Design (RBD), or Risk-Informed Design (RID). These design methods can reduce the effort required to elicit the designer’s true preference by imposing preference models. These preference models, like all models, are abstractions of reality that include some amount of error. Accordingly, they only produce meaningful results if that error is sufficiently small. As such, the value of the design method in a particular design scenario is related to the amount of error between the true preferences and the preference model, as well as the designer's willingness to accept this error.

The following sections seek to justify the application of Utility Theory to the domain of engineering design by examining the methods of RD, RBD, and RID. The limitations that are imposed by the preference models are examined from the context of Utility Theory, and it is shown that the preference models can be replicated within the context of Utility Theory. As a result, it is concluded that Utility Theory is

2.4.1 Robust Design

Robust Design is a method for improving the quality of products and processes by reducing their sensitivity to variations [18-19]. RD is thus a means for reducing risk by reducing the effects of variability without removing the sources of variability. RD is founded on the philosophy of a Japanese industrial consultant, Genichi Taguchi, who proposed that product design is a more effective way to realize robust, high-quality products than by tightly controlling manufacturing processes. Since Taguchi's initial work, many researchers have proposed improvements and modifications to tailor his method to broader engineering applications.

Taguchi's method is based on the Quality Loss Function, which represents Taguchi's philosophy of striving to deliver on-target products and processes rather than those that barely satisfy a corporate limit or tolerance level. The quality loss, L, is proportional by a loss coefficient, k, to the square of the deviation of performance, y, from a target value, T.

$$L = k \cdot (y - T)^2 \qquad \textit{Eqn. (5)}$$

Any deviation from target performance results in a quality loss. This was a departure from common industrial practice in which quality was measured via tolerance ranges. Taguchi's RD approach for parameter design employs designed experiments to evaluate the effect of control factors on nominal response values and sensitivity of responses to variations in uncontrollable noise factors. Product or process designs are selected to maximize the signal to noise ratio, which combines measures of the mean response and the standard deviation. The intent is to minimize performance deviations from target values while simultaneously bringing mean performance on target.

Due to the intellectual and practical appeal of Taguchi's RD philosophy, researchers and practitioners have been actively establishing and improving the methods and techniques needed to implement RD in engineering applications. Many suggestions refer to improvements on statistical and modeling techniques. This area of work falls outside the scope of this thesis; see [20] for an overview. Other researchers have concentrated on the formulation of the objective function in RD. Chen et al. and Bras and Mistree formulate a RD problem as a multi-objective decision using the compromise Decision Support Problem [21-22]. Separate goals of bringing the mean on target and minimizing variation (for each design objective) are included in a goal programming formulation of the objective function. Chen et al. have extended the approach to include alternative formulations of the objective function, such as compromise programming [23] and physical programming [24].

2.4.1.1 Framing Robust Design in a Utility Context

Approaches for RD have in common a general form of the objective function as a weighted sum of mean and variance. This general form is shown in Table 1, here μ is the mean of the objective, σ^2 is the variance of the objective, and α is a positive constant; both the mean and variance depend on the vector X.

Table 1. General Robust Design Optimization Formulation

Find:	$\mathrm{X} = \{\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{N}\}$
That Maximizes:	$f\ (\mathrm{X}) = \mu(\mathrm{X}) - \alpha \cdot \sigma^2(\mathrm{X})$

This formulation reflects a preference for lower variance, which is a form of risk aversion. From a utility perspective, it can be shown that this general RD formulation is equivalent to the maximization of expected utility assuming constant absolute risk aversion and normally distributed utility. Exponential utility reflects constant absolute risk aversion and is shown in Eqn. 6, where v is the value or objective, R is a positive risk aversion parameter, and u is the utility.

$$u(v) = \frac{1 - e^{-Rv}}{R} \qquad \textit{Eqn. (6)}$$

The expectation of this utility, assuming that the objective, v is normally distributed is given in Eqn. 7, where again μ and σ^2 are the mean and variance of the normally distributed objective v.

$$E[u] = \frac{1 - e^{-R\mu + \frac{1}{2}R^2\sigma^2}}{R} \qquad \textit{Eqn. (7)}$$

The transformation $1 - e^{-x}$ is a monotonically increasing function and preserves the maximum as a result. Therefore, the two objective functions are equivalent for the purpose of finding the maximum. Comparing Eqn. (7) to the general RD optimization formulation as described in Table 1, the coefficient α can be characterized by the equation,

$$\alpha = \frac{R}{2}, R \geq 0 \qquad \textit{Eqn. (8)}$$

Because RD formulations are equivalent to assuming constant absolute risk aversion, designers should only use RD formulations when this assumption is appropriate. The assumption of constant absolute risk aversion alone, however, significantly reduces the effort required to elicit a utility function. Three points are needed to fit an exponential utility function, but if the best and worst outcomes are arbitrarily assigned utilities of 1 and 0, respectively, the consideration of a single vN-M lottery is sufficient to characterize R. Therefore, when designers are prepared to assume constant absolute risk aversion, direct elicitation of an exponential utility function is more rigorous than an arbitrary assignment of weighting values and does not require significant additional effort to develop.

When multiple objectives are present, the preference model is necessarily more complicated. Many RD researchers have included tradeoffs between means and variances of multiple objectives in RD formulations using weighted sums. In these cases there is little justification for the weights that are used on each factor. Since it has been demonstrated that the assumption of constant absolute risk aversion significantly reduces the effort required to elicit conditional utility functions, it seems reasonable to suggest that the more rigorous method of preference elicitation in MAUT be applied in the case of multi-objective robust design.

It is important to note that the expectation of the exponential utility function that is equivalent to the general RD formulation assumes a normally distributed objective. This limiting assumption is tied to the use of mean and variance as statistics in robust design formulations. The assumption of constant absolute risk aversion under a utility-based framework does not have this limitation, as the expectation of exponential utility can be computed using sampling procedures for non-normally distributed objectives.

2.4.2 Reliability-Based Design

Based Design is a method that was developed to help designers manage the risk associated with the failure of products. RBD accomplishes this by including direct considerations of an alternative's reliability as part of its evaluation. Rao defines reliability as 'the probability of a device performing its function over a specified period of time and under specified operating conditions' [25]. This definition is also consistent with the expectations of consumers, as they expect any product they purchase to perform its function without failure. Or, they expect to be recompensed if it does fail, unless they were at fault for its failure. Mathematically, the reliability of an alternative X can be defined as,

$$Rel(X) = prob(g(X) < 0) \qquad \textit{Eqn. (9)}$$

where $g(X)$ is the limit-state function that is negative when failure occurs.

RBD can be divided into two different formulations [25]: Reliability-Constrained Optimization (RCO) and Cost-Constrained Optimization (CCO). The formulations are similar in that they both recognize that the designer can manipulate several design variables that in turn have an impact on overall reliability, as well as on cost or other attributes of importance. The key difference between the two is the manner in which they address reliability. RCO treats reliability as a constraint or goal, while maximizing or minimizing some secondary objective. CCO treats reliability as the secondary objective, and instead constrains the cost of the alternative. This distinction is important for the next sections where the preferences are restructured from the perspective of Utility Theory.

2.4.2.1 Framing RCO in a Utility Context

The RCO preference structure will be addressed first, for which the problem statement is shown in Table 2. As shown in the table, RCO seeks to optimize an objective f while maintaining some minimum acceptable system reliability, $\mathrm{Rel}_{\mathrm{crit}}$. Chandu constrained reliability while minimizing the weight of structural supports [26]. Enevoldsen used RCO to minimize total lifetime cost under reliability constraints [27]. Many methods have been proposed to make the solving of reliability constrained problems less computationally expensive, but this is beyond the scope of this thesis. The interested reader is referred to [28-29].

Table 2. RCO Formulation

Find:	$X = \{x_1, \ldots, x_N\}$
That Maximizes:	$\mathbf{f}(X)$
Subject to:	$\mathrm{Rel}(X) \geq \mathrm{Rel}_{crit}$

As illustrated in Figure 5, the reliability constraint is not necessarily always active, depending on the location of the optima of the objective function. For an objective function f2(X) with the original optimum meeting the constraint, the constraint is inactive and the original optimum is maintained. Objective functions such as this are trivial to solve, and therefore they will not be addressed further here. However, for another objective function f1(X) with an optimum not meeting the constraint, the constraint will be active and the optimum as prescribed by the RCO will shift to the constraint, where $X = \widehat{X}$.

Preference models defined using Utility Theory are one-dimensional, and therefore are not suited to handle the lexicographic nature of the constraint on reliability. In order to support a constraint like the one found in RCO, the objective function must be constructed such that the optimal set of parameters automatically results in a system that satisfies the constraint.

In order to further investigate the manner in which RCO (as well as CCO) can be structured within the context of utility, an objective function in terms of the net profit resulting from a particular design alternative X is introduced. In Eqn. (10), $f(X)$ is a deterministic function defining the gross profit of producing and selling the product without failure, C_f is the cost of failure, and $\mathrm{sgn}(\cdot)$ is the signum function. Recall from Eqn. (9) that $g(X)$ is the limit-state function that defines if the product fails. The expected utility of the net profit is calculated by Eqn. (11), which is a function of the reliability.

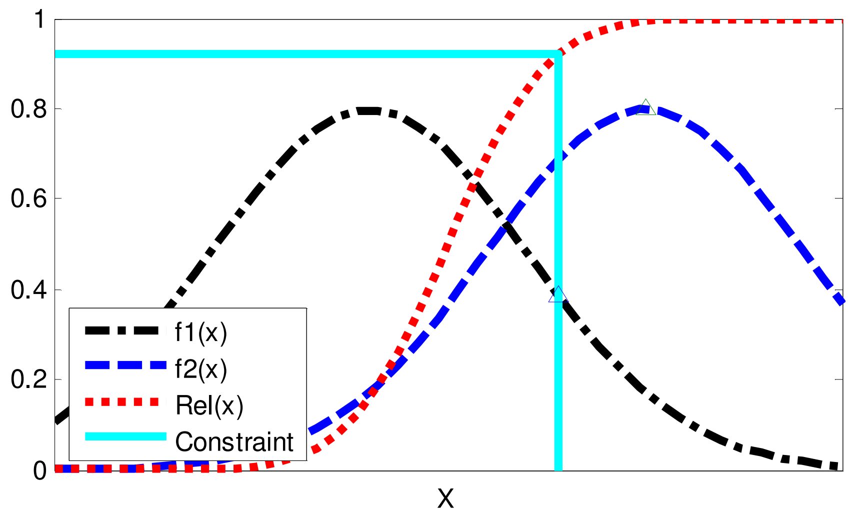


Figure 5. The Reliability Constraint

$$Net\ Profit = \boldsymbol{f}(X) - \left(\frac{1 - sgn\big(g(X)\big)}{2}\right) \cdot C_f \qquad \textit{Eqn. (10)}$$

$$E[U(Net\ Profit)] = Rel(X) \cdot U\big(\boldsymbol{f}(X)\big) - \big(1 - Rel(X)\big) \cdot U(\boldsymbol{f}(X) - C_f) \qquad \textit{Eqn. (11)}$$

Using the single attribute utility function, the optimal design alternative is the one that maximizes the expected utility of the net profit. The designer is free to define the utility function over net profit as described previously, but here the constant risk aversion utility function of Eqn. (6) is referenced for simplicity.

It is conceded that such a utility function may not exactly characterize the utility of profit for the entire design space. However, for a deterministic objective function the utility function and RCO will have identical characterizations of the location of the optimum for certain parameters. The value of the methods arises from their ability to locate the true optima, and therefore the methods' inability to characterize areas of the design space that are not optima is immaterial for deterministic functions. However, if the objective functions are uncertain, the model would need to be accurate in the design space near the optimum as well. As the model's prediction of the true utility becomes less accurate, it may begin to make incorrect predictions about the optimum. In either case, the model's value is that it is capable of predicting the optimal set of parameters. In the remainder of this section a case study is introduced to describe how RCO can be configured to predict the optima as defined by the utility of profit.

2.4.2.2 Case Study-RCO

A designer is sizing a hydraulic cylinder for use in industrial construction equipment. The designer is trying to what thickness X (in meters) that the cylinder casing should be to withstand the pressures generated by the system. From previous experience, he knows that the reliability of the system is directly related to X via the function Rel(X). He also knows that when failure occurs, his company is liable for $C_f = \$3{,}000$, the amount of external damage likely done. He wants to maximize the company's profits $\mathbf{f}(X)$ (in thousands), which are diminished as material costs increase. These relationships are shown in Figure 6.

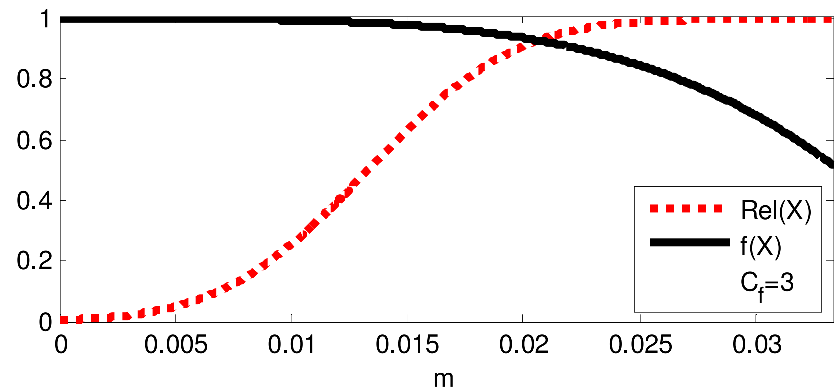


Figure 6. f(X), Rel(X) for RCO Case Study

The designer elicits his risk aversion[2], and finds it to be accurately modeled with a constant risk aversion constant R=0.75. Based on this information, he optimizes Eq. (11) to determine that the optimal thickness is 2.64 cm as shown in Figure 7. Alternatively, the designer could have estimated the required reliability constraint for RCO, and for a constraint of 99.56% reliability, he would have arrived at the same thickness as shown in Figure 7.

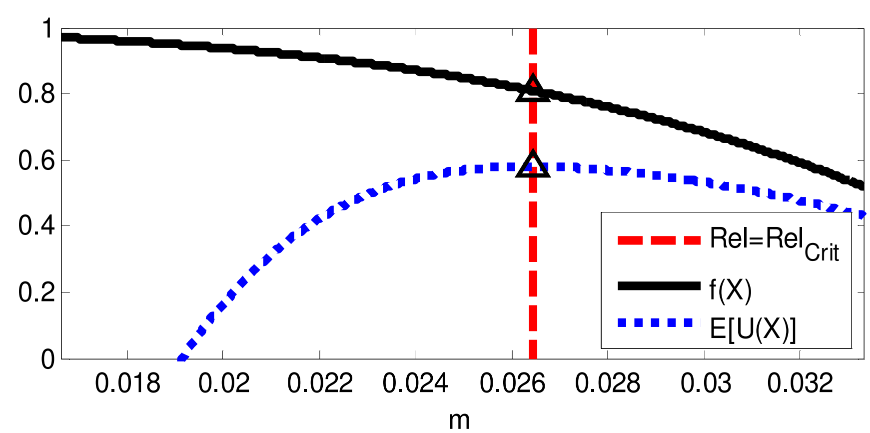


Figure 7. Optima Determination for RCO Case Study

[2] The process of risk aversion elicitation is beyond the scope of this thesis. The interested reader is referred to [15].

Extrapolating the RCO case study, a designer should be able to identify the optimal alternative $\widehat{X}$ for any given $\mathrm{Rel}(X)$, $f(x)$, and C_f, so long as he has identified his risk aversion coefficient R. Likewise, it should be clear that the designer should be able to identify $\widehat{X}$ if he is able to instead identify a proper reliability constraint. However, it may be difficult to meaningfully identify the constraint on reliability. The merits of RCO relative to utility and other methods are further examined in the discussion.

2.4.2.3 Framing CCO in a Utility Context

CCO also contains a constraint, but this formulation instead constrains cost (or some other secondary objective) while maximizing system reliability. Therefore, the objective function requirement is altered slightly, as shown in Table 3, so that cost is the important requirement on the objective function, not reliability.

Table 3. CCO Formulation

Find:	$X = \{x_1, \dots, x_N\}$
That Maximizes:	$R(X)$
Subject to:	$C(X) \leq C_{crit}$

Because of the way it is formulated, CCO tends to be better suited for situations in which designers with a fixed budget need to decide which risk mitigating activities would be the most prudent to perform. Mehr and Tumer introduced the RUBRIC tool which allocates resources based on these principles [10]. Qiu et al. introduced a similar R-DRAM tool that allocates resources in a collaborative and distributed environment [30].

2.4.2.4 Case Study-CCO

If the designer in the previous case study decides to take a CCO perspective, then the designer needs to define an additional function C(X) (in thousands) relating cost to the choice of design alternative. The reliability function and cost of failure remain the same, and the designer's beliefs are shown graphically in Figure 8.

Given that the designer would wish to minimize costs in the design of the hydraulic cylinder the function $\mathbf{f}(\mathrm{X})$ from Eqn. (10) is redefined using Eqn. (12).

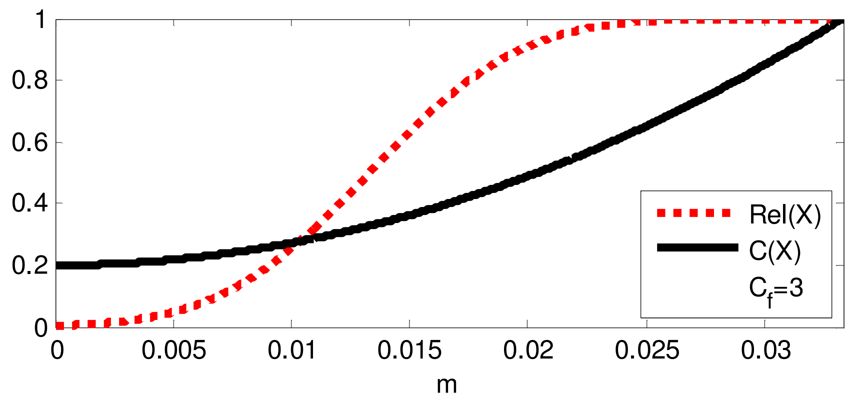


Figure 8. C($\mathbf{X}$), Rel($\mathbf{X}$)for CCO Case Study

$$\boldsymbol{f}(X) = -C(X) \qquad \textit{Eqn. (12)}$$

Then, given the designer's previously elicited risk aversion of 0.75, he could optimize Eq. (11) to find the optimal thickness to be 2.59 cm as shown in Figure 9. **Error! Reference source not found.**Or, the designer could have used the cost constrained formulation and set the arbitrary cost of $6,835 for which he would have reached the same decision about thickness as shown in Figure 9.

Identically to the RCO case study, the designer was able to identify the optimal alternative $\hat{X}$ given only his beliefs about how the reliability and costs of alternatives are related, the costs of failure, and a utility function. Likewise, it should be clear that the designer should be able to identify $\hat{X}$ if he is able to instead identify a proper cost constraint. Once again however, it is difficult to define this constraint because no clear way is provided to meaningfully identify what the upper bound on cost should be. The merits of CCO relative to utility and other methods are further examined in the discussion.

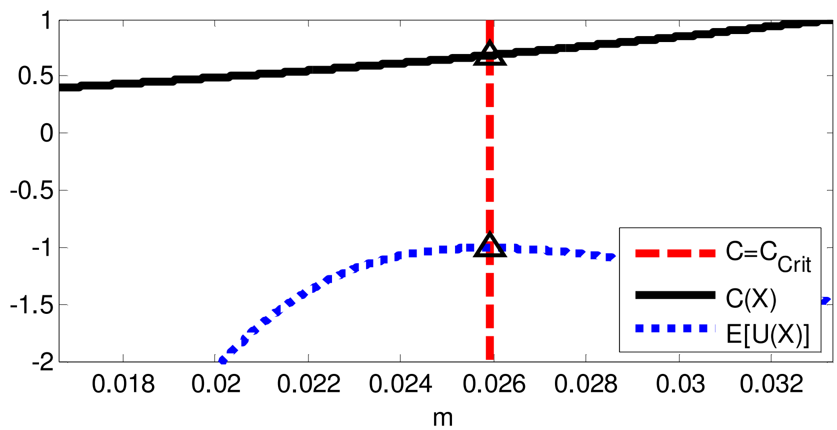


Figure 9. Optima Determination for CCO Case Study

2.4.3 Risk-Informed Design

In this section, another method for design under uncertainty that many have termed Risk-Based Design [10, 31] is addressed. Here, the terminology of Risk-Informed Design is adopted, recognizing that the risk of a design should not be used as the only basis of a decision [32]. Rather, the risk of an alternative is compared to other objectives in a multi-attribute decision problem to determine the optimal set of parameters. Tradeoffs between risk and other attributes such as cost and technical performance are intuitive for users, and is the state of the art in risk management for NASA projects according to [32].

A simple formulation for RID is shown in Table 4, where X are the system parameters for a particular alternative, and $\mathbf{f}(X)$ is the attribute of concern under risk, such as cost, performance, or environmental impact. The RID formulation is built upon the on the idea that the DM is willing to make tradeoffs between the expected value and risk of a design alternative. To that end, RID utilizes the structure of a multi-attribute decision problem in which the designer describes the designer's attitude towards risk by directly describing how important it is relative to the attribute. If $\alpha = 1$ the designer is risk neutral, for $\alpha > 1$ the designer shows risk aversion, and $0 < \alpha < 1$ indicates that the designer is risk seeking.

Table 4. RID Formulation

Find:	$X = \{x_1, \ldots, x_N\}$
That Maximizes:	$\mathbf{f}(X) + \alpha \cdot \mathrm{Risk}(X)$

2.4.3.1 Case Study-RID

Revisiting the case study of the design of a hydraulic cylinder, if the designer instead decided to make the decision of case thickness using RID, he would define the risk as the probability of failure multiplied by the consequences of failure, as shown in Figure 10.

The net profit function $\mathbf{f}(X)$ is identical to the form used for the RCO case study, so the same optimal thickness of 2.64 cm would arise for the designer's risk aversion of

0.75. It can be shown through optimization that the RID formulation would come to the same result for α = 3.62 (See Figure 11).

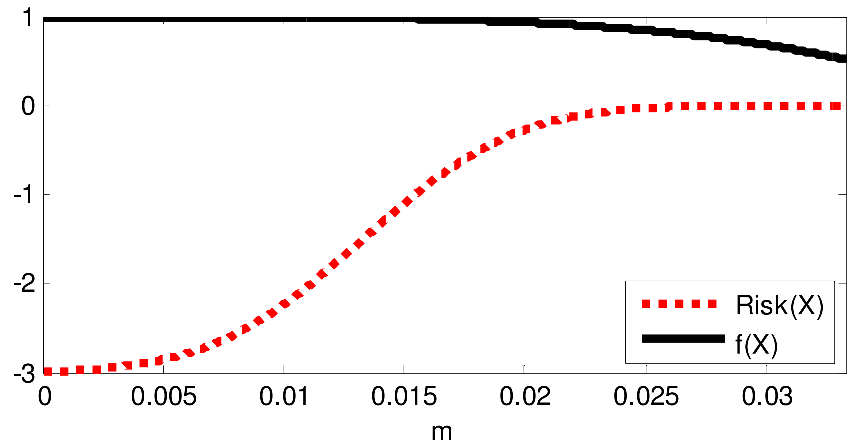


Figure 10. Risk(X), f(X) for RID Case Study

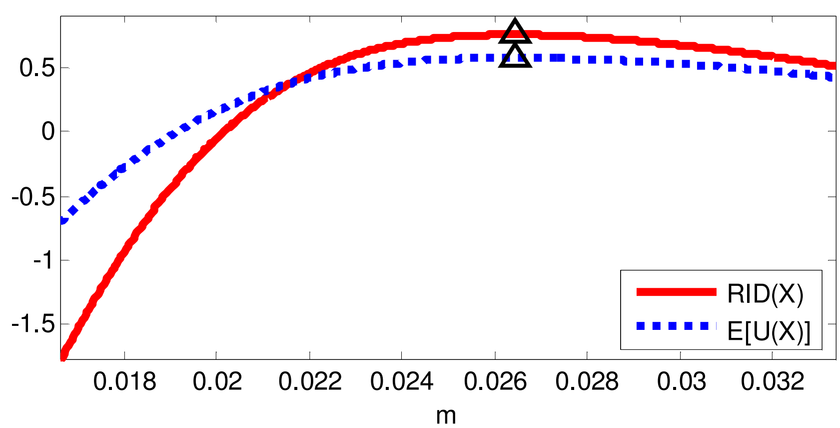


Figure 11. Optima Determination for RID Case Study

Again, the designer would be able to identify the optimal thickness according to his preferences using either method. However, the RID method may be difficult to utilize meaningfully in practice due to the lack of a guideline for determining the value for the coefficient α. In the next section, the methods presented thus far are further investigated.

2.4.4 Comparing the Alternative Methods

The methods for design under uncertainty presented in this work all include underlying preference models of some sort. In the previous sections, these preference models have been identified mathematically, and shown that these preference models can be replicated in terms of Utility Theory. In this section, the methods are further compared based on their costs and merit.

Because utility functions are defined directly from the DM's preferences, he or she can make evaluations using a meaningful basis. On the other hand, each of the methods discussed in this work rely on some arbitrarily chosen coefficient or constraint at some level. In some instances, these constraints or coefficients can be defined heuristically. For example, a designer with a strict budget may find the process of defining a utility function to be unnecessary if she knows that she will need to use all of the available funds. In such an instance, cost-constrained Reliability-Based Design may be appropriate. Or, in other instances, building codes may require that a structure have a minimum reliability. In that case, one would expect reliability-constrained Reliability–Based Design to be the most cost-efficient method. If a quick evaluation of alternatives is required, Risk-Informed Design may be appropriate. However, it may not be appropriate to use of RID for detailed design evaluations, as there is not a meaningful way to determine the coefficient α. It has been shown that Robust Design can be exactly replicated as constant risk aversion with normally distributed uncertainty. As such, Robust Design is suitable for scenarios where such assumptions are close to reality. This is not likely to be the case if there are discontinuities in the objective function, such as those that would arise from failure.

Amongst the individual methods, it is difficult to compare the cost of use at a general level. The amount of resources each method would require is likely to vary significantly based on the particular design scenario. Due to its complexity, Utility Theory is likely to be the method that requires the greatest amount of effort; the careful elicitation of preference is a tedious process that requires a solid understanding of the underlying mathematics. The other methods are likely to have costs that are roughly similar to each other, as they rely on similar inputs from the designer. For example, an essential task for the methods is the determination of product reliability. The calculation of reliability can be quite time-intensive, or may require expert knowledge. However, this cost is common between the methods, as each method requires knowledge about the possibility of failure. As such, the cost of computing reliability should not differentiate the methods. One differentiating aspect is that RBD involves constrained optimization, which can be more complex to solve than unconstrained optimization. In general, this may make RBD more computationally expensive to evaluate than the other methods.

A final attribute of importance in the comparison of these methods is their robustness to errors in the elicitation of preference. If a designer elicits his or her preference, it is possible that the value of the parameter he or she elicits could be slightly different from the value that best reflects his or her true preference. It is therefore desirable for the methods to be robust against small variations in designer input. Because Robust Design is well-suited only for cases in which a well-behaved objective function exists, it is not expected that small changes in α would result in large changes to the optima. Similarly, the coefficient in RID has a linear relationship with risk. Therefore it is not expected that small variations would result in large shifts in the optima for RID either. RBD, on the other hand, uses constraints on either reliability or cost to define that which is acceptable. When RCO is used, reliability tends to be constrained to relatively high levels. At such levels, the cost of achieving an additional unit of reliability can increase drastically. As a result, the optimal alternative may shift drastically as well. The same is not necessarily true for CCO, as small changes in the cost constraint tend to result in only small changes to reliability.

2.5 Summary

In this chapter, it has been shown that engineering design is a suitable domain for the application of Utility Theory. A foundation of Utility Theory has been reviewed and examined. It was noted that Utility Theory is not applied to engineering design in widespread fashion, as the difficulty in eliciting preference is viewed as an imposing obstacle. However, in this Chapter, Utility Theory was utilized to examine the limitations on preference imposed by the methods of Robust Design, Reliability-Based Design, and Risk-Informed Design. These methods have been used for design under uncertainty much more commonly than Utility Theory. It was that these methods are suitable surrogate models for a designer’s true preferences under certain assumptions, and these assumptions were investigated from the perspective of Utility Theory.

CHAPTER 3

THE TEMPORAL ANALYSIS DECISION MODEL

3.1 A Review of Relevant Literature

In this Chapter, the Temporal Analysis Decision Model (TADM) is presented as a tool which can be leveraged to analyze engineering design case studies. In this section, previous work related to the analysis of design decisions and the impact of temporal costs on net utility is reviewed.

In [1], Thompson and Paredis investigate the value of a process-centric problem formulation and identify a need for consideration of the costs of the design process in decision making. They present a decision model which considers the monetary costs of the design process in design decisions. The model is also compared to current Value of Information work, and is found to provide better information when a sequence of tests is available. However, they do not include temporal costs in the decision model, and as such, they are unable to examine how the prescribed behavior of a designer depends upon temporal degradation of product utility.

Motte [33] presents a similar decision model in which he examines the utility of three different strategies for handling the uncertainty prevalent in the design phase. In his simple decision model, Motte identifies several direct monetary costs of refining design alternatives as well as costs due to time. However, he accounts for time purely based upon linear cost per unit time based largely upon wages or equipment costs, and neglects the possibility of temporal degradation of product utility. Also, his model accounts only for discrete outcomes, and is specialized such that it cannot be generalized to model a generic design decision.

In [2], Hazelrigg argues that in an enterprise context the net profit should be the primary driver of utility, and recognized the breakdown of profit into revenues and costs. He also stated that time has an important value in these decisions and reinforces his argument with an example about interest and discount rates. However, his definition of the value of time focuses primarily on delay in revenue streams. In the context of business and marketing, time can often be valuable in other ways.

Pawar et al. [34] argue that design process time is valuable in that it directly affects the Time to Market (TTM) of a product. TTM is a measure of how long it takes for a product to reach the marketplace. According to Pawar, gross profit is strongly affected by TTM, as products with shorter total TTM's reap the benefits of extra sales revenue, earlier breakeven on investments, extended sales life, and increased market share.

Urban et al. [35] also stresses the importance of a short TTM and early product launch under competition. They developed a simple model that demonstrates the market advantage granted to pioneering products. From the model, the earliest entrant to a market earns the largest market share, with later entrants being forced to produce superior goods, advertise heavily, or cut prices in order to gain market share themselves.

Reinertsen [36] investigated the effect of a six month launch delay on cumulative profit for a product in which market price decayed each year. He found that the rate of price decay greatly impacts the profitability of the product, quickly reducing the total profit by 100%. Reinertsen also made the dramatic claim that, for a product with a five year lifespan, 'six months delay can be worth 33 percent of life cycle profits' [37]. The incentive to curtail such significant losses should be a strong driver to alter designer behavior during the design process, especially since 80-90% of the TTM equation is determined in the design phase [38]. Based on this information, it seems proper that engineers should consider the impacts of their decisions on TTM, and make tradeoffs accordingly. The next section introduces the TADM, a decision model that can be used examine these impacts and tradeoffs.

3.2 Introducing the Temporal Analysis Decision Model

In the decision model proposed by Thompson and Paredis [1], the design decisions available to the DM are to either 'Analyze' or 'Select' one of two alternatives. The option of 'Analyze' is used to describe any of several methods of gathering additional information about a design alternative. The analysis could refer to marketing analyses, prototype development, or computer simulation. The important aspects of an analysis in the model are (1) that the analysis provides information about the alternative tested which can be used to reduce uncertainty, and (2) that the analysis consumes some resources. The option of 'Select' is used to describe the process of refining the focus in the design process. In some contexts, 'Select' could mean that a final design is determined and finalized plans are sent to be manufactured. Another viewpoint is that a 'Select' decision is merely a decision to perform further refinement on that alternative only.

In Figure 12, a decision model is presented that is similar to the model proposed by Thompson and Paredis. The decision model is shown in the form of a decision tree, which is a method of visualizing decisions. In the new decision model, the utility function accounts for the duration of analysis by modeling its effect on value. A second difference is that a new decision alternative is introduced, ‘Analyze A and B’. The ability to analyze both alternatives immediately is significant, because its availability acknowledges the importance of parallel analysis when time is important.

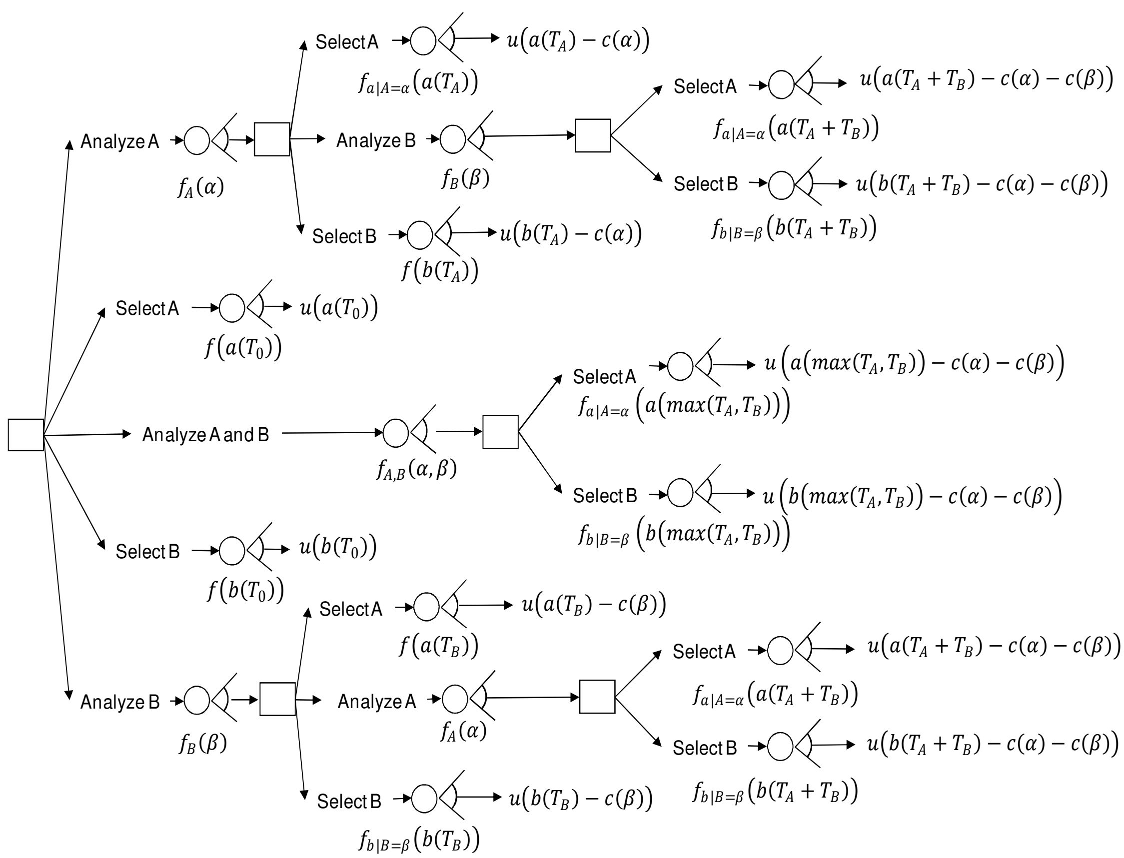


Figure 12. Decision Tree for Modified Decision Model

In a decision tree, boxes represent decisions and circles represent chance events. Arcs emanating from decision boxes represent decision alternatives, whereas arcs emanating from chance events represent outcomes. Performing an analysis on one of the products enables the DM to update his prior knowledge about that product based on the outcome of the analysis. In Bayesian terms the prior knowledge is represented as the prior distribution and the updated knowledge is defined as the posterior distribution. In this decision tree, a(t) and b(t) represent the earnings of products A and B if released at time t. The estimated values from analyses on A and B are represented by α and β, respectively. The functions under the chance events represent probability density functions, where

- $f(a(t)\,)$ and $f(b(t))$ are the prior distributions of the earnings of products A and B at time t, respectively.
- $f\_A\ (\alpha)$ and $f\_B\ (\beta)$ are the marginal distributions of the outcomes of the analyses on products A and B.
- $f\_(A,B)\ (\alpha,\beta)$ is the joint marginal distribution of the outcomes of the analyses on both products A and B.
- $f\_(a|A=\alpha)\ (a(t))$ and $\mathrm{f}\_(b|B=\beta)\ (b(t))$ are the posterior distributions of the earnings of products A and B given the values from the marketing analysis.
- $u(*)$ is the utility of the argument.
- $c(*)$ is the monetary cost of performing the analysis.
- $\mathbf{T_A}$ and $\mathbf{T_B}$ are the amount of time required to perform the analyses on A and B.

According to Utility Theory, the DM should select the alternative with the largest expected utility. For a decision tree, a DM calculates the expected utility by rolling back the branches. At decision nodes, a rational decision maker will always select the best option, and therefore the maximum utility of all alternatives at that node is rolled back. At chance nodes, the branch is rolled back by calculating the expectation over all possible outcomes of the event [39].

3.3 Introducing Some Basic Simplifying Assumptions

The equations generated by evaluating the decision tree are generalized such that any formulation of uncertainty can be represented. Therefore, the first step required to solve the model is to formalize the parameters mathematically. In this thesis, normal probability distributions are used to model the uncertainty about performance. The normal distribution was chosen because it simplifies the calculation of the posterior distributions conditional on the outcome of analyses. An example for two products is shown in Figure 13. As shown in the figure, the expected net profit of A is slightly less than that of B, but B is much more variable. This introduces an overlap region in which A could produce greater profits than B.

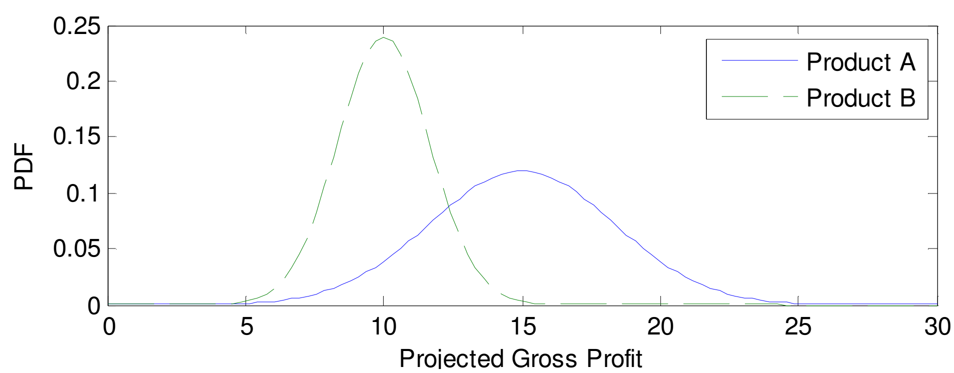


Figure 13. PDFs for Products A and B

The analyses in the first scenario have three important characteristics: Cost, Quality, and Time. The monetary cost is defined in dollars and directly affects the net profit. The quality is defined by how accurately the actual value can be predicted. Mathematically, it is assumed that the result of an analysis is the true result, plus a random error term. It is further assumed that the error term is unbiased and normally distributed as well. The relationship is expressed in Eqn. (13).

$$v_a = v + \varepsilon \, , \varepsilon \sim N(0, \sigma_\varepsilon^2) \qquad \textit{Eqn. (13)}$$

where v_a is the predicted gross profit from the analysis, v is the true earnings, ε is the random error term, and σ_ε is the standard deviation of the error term.

The time required for an analysis to be performed can be quite valuable if the expected gross profit changes due to product development delays. In Figure 13, the designer does not expect the gross profit to change at all as long as a product is selected before the deadline. However, if no product has been selected when the deadline is reached, then the expected gross profit of both products will be zero, with any costs incurred reducing the net profit. To model the time varying behavior of gross profit, the following formulation is proposed,

$$v(t) = v_0 \cdot c(t) \qquad \textit{Eqn. (14)}$$

where v(t) is the projected gross profit at time t, v_0 is the initial projected gross profit, and c(t) is a scalar function that models the DM's beliefs about how gross profit changes as time progresses. c(t) can be any scalar function so long as it is defined over the entire range of possible times. When Eqn. (14) is combined with the assumption that uncertainty is normally distributed, the mean and standard deviation of the gross profit as a function of time can be defined as,

$$\mu(t) = \mu_0 \cdot c(t) \qquad \textit{Eqn. (15)}$$

$$\sigma(t) = \sigma_0 \cdot c(t) \qquad \textit{Eqn. (16)}$$

For a simple scenario involving a firm deadline, c(t) can be modeled as a step down function as shown in Figure 14. The effect of a step down function is that the mean and standard deviation remain at the initial nominal levels until the deadline is reached, at which time the mean and standard deviation both become zero. The formulation results in value being deterministically zero after the deadline, as expected.

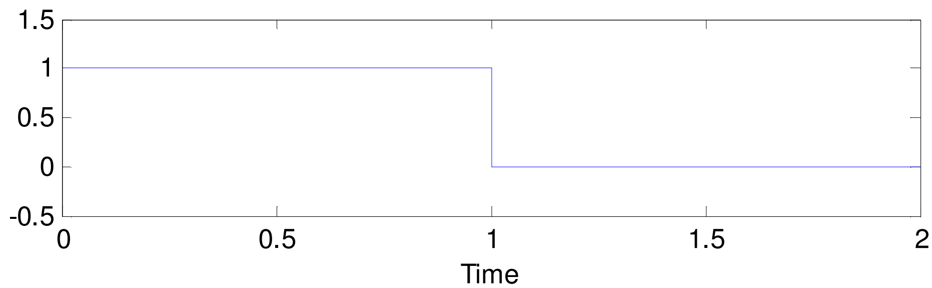


Figure 14. c(t) for a Simple Deadline Scenario

Bayes' theorem is used to compute the posterior probabilities of the gross profits of products A and B given the results of the analyses. The properties of the prior normal distribution are such that the posterior distribution is also normally distributed. The derived mean and standard deviation of the posterior probability are shown in Eqn. (17).

$$f_{a|A=\alpha}(v_\alpha, t) \sim N\left(\frac{\sigma_a(t)^2 \cdot v_\alpha + \sigma_\varepsilon^2 \cdot \mu_a(t)}{\sigma_a(t)^2 + \sigma_\varepsilon^2}, \sqrt{\frac{\sigma_a(t)^2 \cdot \sigma_\varepsilon^2}{\sigma_a(t)^2 + \sigma_\varepsilon^2}}\right) \qquad \textit{Eqn. (17)}$$

where $\mu_a(t)$ and $\sigma_a(t)^2$ are the mean and variance of the prior distribution of the gross profit of Product A after time t, respectively.

When faced with decisions with uncertain outcomes, DM's may not always make decisions purely based upon the expected value. The amount of variance around each mean is also usually important information. If a DM is willing to give up some amount of performance in order to reduce the amount of uncertainty about an outcome, he is said to be risk averse. If the DM does not take variance into consideration, he is said to be risk neutral. In this thesis, it is assumed that the DM has a constant risk aversion R, in the form of Eqn. (6). For R equal to zero, the DM is risk neutral, and for R increasing above zero, the DM is increasingly risk averse. With a normally distributed value and an exponential utility function, the expected utility is defined by Eqn. (7), where μ_v and σ_v^2 are the mean and variance of the value distribution.

With all the parameters of the model thus defined, the expected utility of each decision alternative can be calculated. In the next section, both scenarios are examined using the model, and the optimal decisions are compared to those predicted by the model when the effects of time are not considered.

3.4 Investigating the Decision Model Parameters

3.4.1 Design Alternative Parameters

As described above, the assumption of normally distributed uncertainty has been imposed on prior distributions for this thesis. It should be noted that this assumption is not necessary, and does not limit the capability of the decision model to handle situations when uncertainty is not normally distributed. This assumption is only imposed because it is likely to be reasonable, and because it allows for significant simplification in evaluating the model. It allows the use of Eqn. (17) to define the posterior distribution in closed form. It also allows the simple calculation of the distribution of utility at any time using only a coefficient multiplied by the mean and standard deviation.

However, if normally distributed uncertainty is not a reasonable assumption for a particular scenario, the model could instead be evaluated using Monte Carlo simulation or some other sampling based method. However, as the Temporal Analysis Decision Model is not the main contribution of this thesis, but rather serves as a tool, further comments sampling based methods are not included in this thesis.

3.4.2 Analysis Parameters

Analyses are also assumed to have a certain amount of normally distributed "noise" around an unbiased mean prediction. As described above, the normal distribution of noise is used as a simplifying assumption, but its use is not necessary to evaluate the decision model.

Secondly, the assumption has been introduced of certain costs, durations, and qualities of analyses. In reality, this may not be appropriate. It is likely to be the case that the designer is uncertain about exactly how long an analysis will be, or how much it will cost. However, these assumptions seem reasonable to the degree that the designer should be able to determine an expectation on these parameters. Additionally, it is largely unexpected investigating the variations in these parameters would lead to additional findings regarding the hypotheses. As such, the investigation of uncertainty in these parameters is reserved for future work.

3.5 Scaling Considerations

The parameters for the distribution of projected gross profit are normalized such that the alternative with the largest standard deviation has a mean of zero and standard deviation of one. Both distributions, as well as the parameters of the analyses are normalized, and a zero calibration value is introduced to adjust for the lateral shift in the means of the distributions. The zero calibration value (ZCV) is determined by Eqn. (18) and is depicted graphically in Figure 15. Zero Calibration Value The ZCV is important because it accounts for changes from normalization and allows c(t) to affect value in the same way as it would a non-normalized distribution. The modified calculation for the mean of a normalized distribution as a function of time is given in Eqn. (19).

$$ZCV = -\frac{\mu_B}{\sigma_B} \qquad \textit{Eqn. (18)}$$

$$\mu(t) = \mu_N \cdot c(t) + ZCV \cdot \big(1 - c(t)\big) \qquad \textit{Eqn. (19)}$$

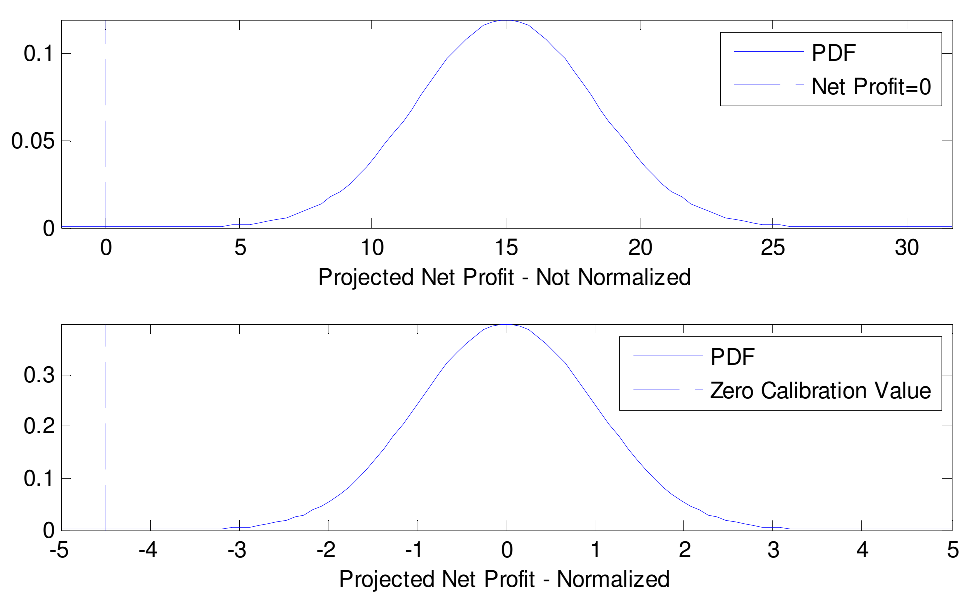


Figure 15. Zero Calibration Value

3.6 Summary

In this Chapter, the Temporal Analysis Decision Model, which contains the actions available to designers (Analyze, Select) as decision alternatives, was presented. The relevant literature was reviewed. A model for estimating product utility degradation due to temporal passage was also introduced. Assumptions that simplify the computational evaluation of the decision model were also introduced and briefly examined. In the next Chapter, the Temporal Analysis Decision Model will be applied two case studies: an OEM parts supplier facing a deadline, and a consumer electronics corporation in a competitive marketplace.

CHAPTER 4

ANALYZING DESIGN SCENARIOS USING THE TEMPORAL ANALYSIS DECISION MODEL

4.1 Leveraging the Decision Model to Analyze Design Scenarios

In this Chapter, the Temporal Analysis Decision Model is leveraged to analyze the prescriptive behavior of designers in two case studies. After the case studies are analyzed, the effects of varying model parameters are investigated via the use of boundary plots.

4.1.1 OEM Parts Supplier

The OEM Parts Supplier case study is a commonly occurring decision in engineering design under time constraints. Quite often, deadlines can arise such that a decision must be made while there is still significant uncertainty about outcomes. In this case, the manager must decide quickly because the automobile manufacturer will not accept bids after the deadline. In practice, these deadlines may be arbitrarily declared, and some flex time may actually exist. However, there are also many situations in which the deadline is firm. For example, NASA space launches must meet their launch windows, else the mission will not be given clearance to take off and the project may be delayed significantly. The OEM Parts Supplier case study is based upon the following problem:

An OEM parts supplier is preparing a bid for a contract to design hydrogen fuel cells for a major automobile manufacturer. The supplier's researchers have developed two new technologies that are expected to outperform the technology of the automobile manufacturer's current supplier. The first technology appears to be rather robust, and is projected to bring in a gross profit of $10M plus or minus $5M. The second technology has the potential of longer life, but the research department is less certain about its robustness in application. The gross profit for the second technology is estimated at $15M plus or minus $10M.

The head designer on the team has six months to choose which new technology to propose before the bid is due. As long as she submits the proposal before the deadline, she is sure that her design will be accepted. However, if she does not submit the proposal on time, the automobile manufacturer will renew its contract with its current supplier and the OEM parts supplier will not get a contract. During the remaining six months, she could have the research department perform further analysis on the technologies. Each analysis would cost $100K, but would also take 4 months to perform, leaving 2 months for design refinement. From previous experience, the head designer knows that the research tests usually give her enough information to predict the actual gross profit within $1.5M. The head designer must decide whether to perform testing to gather more information, or to save the testing costs and select a design for refinement now.

The head designer in the OEM case study is faced with a decision about whether to perform analyses on two design alternatives, or to select one for refinement immediately. The important parameters of the decision, before and after normalization, are summarized in Table 5.

Table 5. Summary of OEM Case Study Parameters

		Mean	Margin	Standard Deviation
Product Prior Distributions	A ($M)	10	5	1.67
	B ($M)	15	10	3.33
Analysis Parameters	Cost ($M)	0.1	-	-
	Quality ($M)	-	1.5	0.50
	Time (% of Deadline)	0.67	-	-
Normalize via σ_B and μ_B				
Product Prior Distributions	A ($M)	-1.5	1.5	0.50
	B ($M)	0	3	1.00
Analysis Parameters	Cost ($M)	0.03	-	-
	Quality ($M)	-	0.45	0.15
	Time (% of Deadline)	0.67	-	-
ZCV	($M)	-4.5	-	-

The expected utility of the net profit for each decision alternative is calculated for a risk neutral DM and risk averse DM with constant risk aversion R=1 (see Eqn. (6)). The expectations of the 'Analyze' decisions are calculated via adaptive Simpson's quadrature, and the expected utility for the 'Select' decisions are calculated via Eqn. (7). In order to compare the prescribed behaviors for a designer when product utility is and is not time sensitive, the Temporal Analysis Decision Model is evaluated for two versions of $c(t)$. For the scenario when product utility is time sensitive, $c(t)$ is defined as a step-down function, as shown in Figure 16. For scenarios when product utility is not time sensitive, $c(t)$ is defined as equal to one.

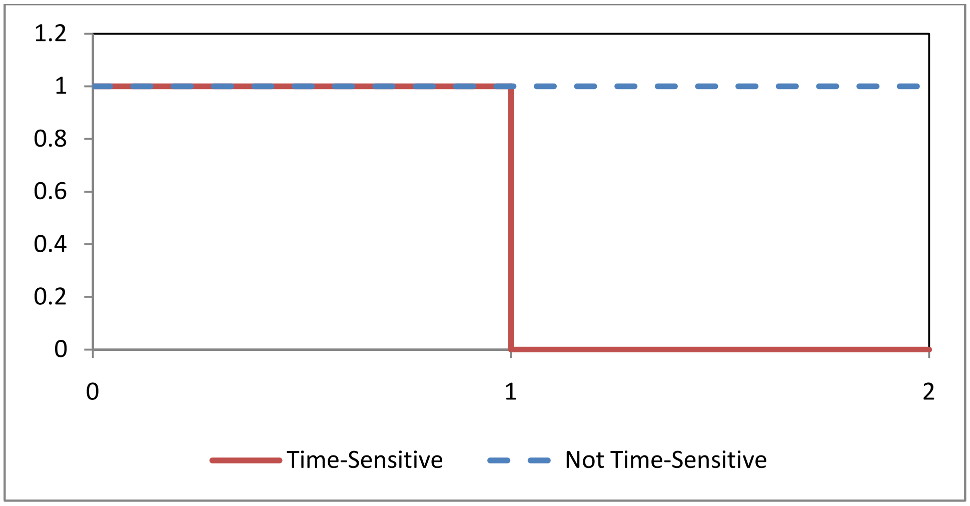


Figure 16. c(t) for the OEM Parts Supplier Case Study

The resulting expected utilities are included in Table 6. The columns labeled "Time" correspond to the case when product utility degrades temporally because of the deadline. The columns labeled "No Time" correspond to a case when the deadline is neglected and product utility does not degrade temporally. The decision alternative with the highest expected utility for each case is shown highlighted and in bold font.

Table 6. Expected Utilities of Decision Alternatives in $M for OEM Case Study

	Risk Neutral R=0 (1/$M)		**Risk Averse R=0.02 (1/$M)**	
	Deadline	**No Deadline**	**Deadline**	**No Deadline**
Select A ($M)	10.000	10.000	9.019	9.019
Select B ($M)	**15.000**	15.000	12.838	12.838
Analyze A ($M)	14.900	14.973	12.685	13.314
Analyze B ($M)	14.993	**15.023**	13.248	**13.388**
Analyze A&B ($M)	14.945	14.945	**13.312**	13.312

As shown in the table, a risk neutral designer that believes the deadline is not important should take the time to *Analyze B*. By looking deeper in the model, it can be shown that this action is prescribed because the designer believes that she would also have sufficient time to perform a secondary analysis on product A. Once temporal degradation of product utility is included, the designer realizes that a second analysis would take too long to perform, and would result in the OEM missing the deadline. As a result, a time-conscious designer should recommend that the resources required for the first test not be wasted, and should therefore *Select B* from the start.

Again, a risk averse manager that does not believe product utility is time-sensitive should perform the action of *Analyze B*. This action is prescribed because product utility is insensitive to time, meaning that the designer has the time to perform one analysis, then decide whether the second analysis is worth performing. However, when faced with a deadline, the designer from the OEM parts supplier does not have the ability to perform tests sequentially. As such, she should decide to analyze both alternatives in parallel, or perform the action *Analyze A&B*.

This simple case study begins to provide empirical justification for the hypotheses. The first hypothesis,

H1: *When considering the temporal degradation of product utility, the maximization of expected utility leads to the parallelization of design tasks.*

is empirically supported by the risk averse designer in this case study. Clearly, a risk averse designer prefers alternatives with reduced uncertainties, and is willing to make tradeoffs with mean performance in order to obtain it. In this scenario, the designer is only able to reduce the uncertainty in both alternatives by testing in parallel. Therefore, if reduction in uncertainty is sufficiently important to the designer that both alternatives need to be analyzed, then it will have to be done in parallel. The second hypothesis,

H2:	*When considering the temporal degradation of product utility, the maximization of expected utility leads to risk acceptance rather than risk mitigation*

is empirically supported by the risk neutral case. In the case study, when the designer is not concerned with degrading product utility, the prescribed action is to *Analyze B*. Once the designer considers the deadline, however, the temporal costs of design increase and expected utilities of the actions change. As a result, the prescribed behavior changes from risk mitigation through uncertainty minimization to risk acceptance through product design alternative selection. The hypotheses are further supported in following sections, which include a second case study and a general exploration of decision model parameter effects.

4.1.2 Consumer Electronics Company

Many design decisions do not have firm deadlines by which they must be made. In this case study, a situation is examined where the DM is forced to decide not by some outside constraints, but by consideration of his own utility. Whereas the OEM case study examines the value of the decision model under constrained circumstances, this case study illustrates the decision model's value in situations where the DM is not constrained by deadlines. The Consumer Electronics Company case study is based upon the following problem:

An enterprise that manufactures consumer electronics is developing its next generation of mobile entertainment devices. The design team has proposed two new product ideas that it feels will be successful in the market. The first product combines the functionality of two older products, while the second product possesses a degree of portability that is currently unrivaled in the marketplace. The VP of New Product Development is relatively confident that the first product will be a success, and the gross profit over its lifetime is projected to be $20M plus or minus $7M. The VP is much less confident about the second product, which has a gross profit projected to be $30M plus or minus $15M.

The marketing division has offered to analyze the products by performing consumer surveys. The analysis would cost $80K per test, and takes it 4 months to perform the tests and compile the results. The marketing division tells the VP that their analyses typically predict the actual profitability of a product within $1M.

In this scenario, there is no deadline for completion that restricts the VP's decisions, but every day the product is still being designed is a day that it is not being sold. The VP must decide whether the analyses are worth performing, or whether it is better to make the decision now and send the product to the market faster. It is assumed that the VP does not want the two products to compete against each other and as such will only select one product for development.

The Consumer Electronics Company (CEC) case study concerns a manufacturing firm faced with a decision about which mobile entertainment device it should develop. The DM in this scenario is the VP of New Product Development. He must decide whether or not to perform a marketing analysis on one of two possible products, which are termed A and B. The important parameters of the decision, before and after normalization are summarized in Table 7.

Table 7. Summary of CEC Case Study Parameters

		Mean	Margin	Standard Deviation
Product Prior Distributions	A ($M)	20	7	2.33
	B ($M)	25	15	5.00
Analysis Parameters	Cost ($M)	0.08	-	-
	Quality ($M)	-	1	0.33
	Time (Years)	0.33	-	-
Normalize via σ_B and μ_B				
Product Prior Distributions	A ($M)	-1	1.4	0.47
	B ($M)	0	3	1.00
Analysis Parameters	Cost ($M)	0.016	-	-
	Quality ($M)	-	0.2	0.07
	Time (Years)	0.33	-	-
ZCV	($M)	-5	-	-

In this case study, the DM is not faced with a deadline, but he knows that the longer it takes to finalize the design, the longer it will take to send the products to the market. During this time, several factors will contribute to the reduction in the gross profitability of the products. Competitors will release rival products, reducing possible market share. In a volatile market like consumer electronics, prices tend to drop quickly, reducing the margin on each product. Demand for the product also tends to change with time. For this case study, a simple model is introduced that includes these various factors and predicts how gross profit is affected by release time.

First, it is assumed that the product has a five year lifespan. During these five years, market price decays at 15% per year. Furthermore, production cost decays at 10% per year to a final non-zero value as manufacturing processes become more efficient. These are average values for consumer electronics products, according to ranges found in [36]. The equations used for market price and production cost are given as Eqn. (20) and Eqn. (21), where starting price is normalized to 1, and t is in years. Market demand, D, for a product over time is modeled as a normal distribution with mean of 2 and standard deviation of 1. It is assumed that the peak of demand occurs at year 2 for this scenario, as it takes time for the product to gain exposure and acceptance in the market. The model for demand also captures the two stages of product life in the market: market growth when the product is being introduced to new consumers, and decay after it begins to become obsolete and is replaced by competition. The functions are plotted in Figure 17.

$$Price(t) = e^{-.1625 \cdot t} \qquad \textit{Eqn. (20)}$$

$$Cost(t) = 0.2 + 0.4 \cdot e^{-.1054 \cdot t} \qquad \textit{Eqn. (21)}$$

$$Profit_{Gross}(T_R) = \int_{T_R}^{5} \big(Price(t) - Cost(t)\big) \cdot D(t)dt \qquad \textit{Eqn. (22)}$$

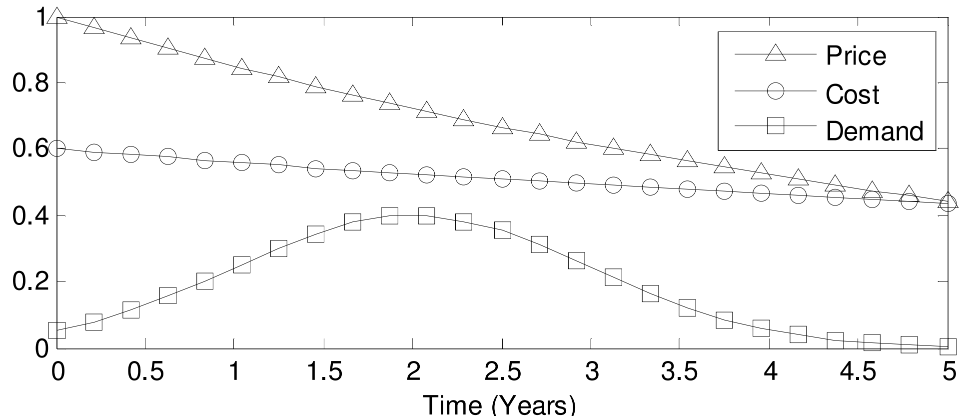


Figure 17. Price, Cost, and Demand for CEC Case Study

In this simple model, the gross profit of a product is formulated as a time integral of price, cost, and demand, as shown Eqn. (22) where T_R is the time that the product is released into the market. The gross profit as plotted in Figure 18 represents how the DM believes the value of an artifact changes as a function of release time. This is exactly the goal of the function $c(t)$ as described previously. For use in this model, Eqn. (22) is normalized to have a maximum value of one, and use the normalized function for calculations in this model. The normalized gross profit, which acts as $c(t)$, is also plotted in Figure 18.

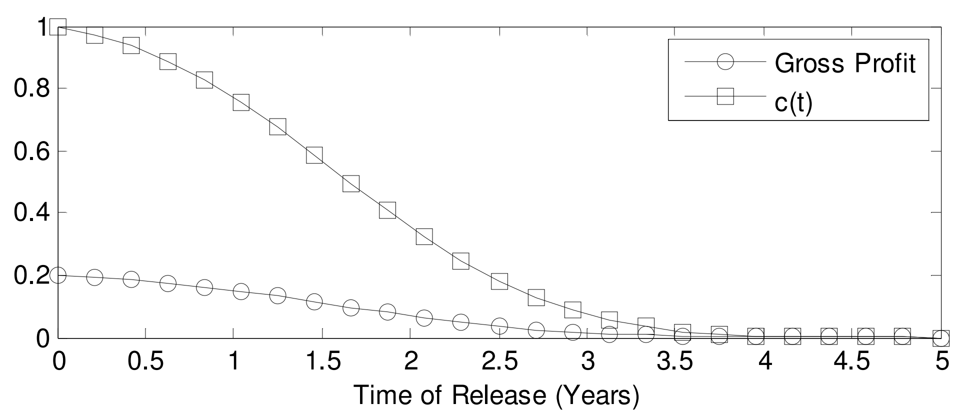


Figure 18. Gross Profit as a Function of Release Time

As seen in the plot, the amount of possible gross profit decays significantly in a short period of time, reaching a gross profit of roughly zero after a delay of about three years. In the context of a competitive marketplace, this appears to make sense. After a market has existed for an extended amount of time, the competition will have gained the advantages of market share and customer loyalty. Therefore, it can become increasingly difficult for a new competitor to enter the market profitably. This simple model for gross profit is not likely to be accurate in most situations. However, this formulation is merely a tool used to understand prescribed behavior under circumstances when product utility is temporally dependent. Therefore, it is not necessary that this model be entirely accurate. It is only required that it provide a reasonable approximation of possible circumstances.

With the new definition of $c(t)$ above, the decision problem can be solved for the CEC case study. The decision tree was evaluated for the parameters in Table 7, and the results are shown in Table 8.

Table 8. Expected Utilities of Decision Alternatives in $M for CEC Case Study

	Risk Neutral R=0 (1/$M)		Risk Averse R=0.01 (1/$M)	
	Competition	**No Competition**	**Competition**	**No Competition**
Select A ($M)	20.000	20.000	17.929	17.929
Select B ($M)	**25.000**	25.000	21.756	21.756
Analyze A ($M)	23.725	25.386	20.118	23.620
Analyze B ($M)	24.106	**25.431**	21.794	**23.654**
Analyze A&B ($M)	24.150	25.384	**22.060**	23.620

This scenario produces similar results to those of the OEM case study. In a non-competitive marketplace (ie. product utility does not degrade temporally) the decision model shows a risk neutral DM should decide that performing the marketing analysis on Product B is the best decision. If the DM were to consider a competitive marketplace, he would see that the time required for the analysis is too valuable, and should therefore decide to *Select B* immediately. Similarly, a risk averse DM that in a non-competitive marketplace should also perform the marketing analysis on B, as he should be willing to pay a premium to reduce the uncertainty about Product B's gross profit. The decision model further prescribes that, in a competitive marketplace, performing the marketing analyses on both products from the start is the best alternative. By testing both alternatives immediately, the DM saves the testing time required for sequential testing, and therefore the amount of profitable time remaining for the product in the marketplace.

The CEC case study provides additional empirical support for the hypotheses similar to the OEM case study. Recall that the hypotheses assert that parallelization of analysis testing and risk acceptance are trends that will result from increasing time-based costs of design process activities. Upon examining the risk neutral and risk averse scenarios of the CEC case study, it is clear that both of these possibilities occur. Similarly to the OEM case study, the risk neutral designer should change from a behavior of risk mitigation (*Analyze B*) to a behavior of risk acceptance (*Select B*). The risk averse designer in the case study instead shifts from performing analyses in sequence (*Analyze B* then *Analyze A*) to performing them in parallel (*Analyze A&B*).

The existence of empirical evidence based upon two case studies cannot prove the validity of a hypothesis, and it is not the intent of this thesis to rely on these case studies as such. However, these case studies do suggest that the hypothesis may be correct, and at least do not refute them. To investigate the trends resulting from product utility being increasingly sensitive to temporal degradation, the decision model is further investigated in the next method through the use of boundary plots.

4.2 Further Exploring Model Parameter Effects Using Boundary Plots

The two case studies examined in this thesis begin to provide evidence in support of the hypotheses. However, it is naive to expect that all design cases should match these exact problem circumstances. Therefore, it is proper to examine how the parameters of a design scenario affect the results of the model. One method for organizing and visualizing the results of the decision model for multiple parameters is a boundary plot. In a boundary plot, the optimal decision for a set of parameters is plotted, where regions with the same optimal decision are grouped by color or shading. In this section, the effects of variations in the parameters of the Temporal Analysis Decision Model are investigated by changing one parameter at a time and examining the changes in the resulting boundary plot. In each boundary plot, the analysis costs, qualities, and times, and ZCV are held constant, while the normalized mean and standard deviation of alternative A are varied. The x-axis of the plots corresponds to the normalized mean of A relative to B, with the y-axis corresponding to the normalized standard deviation of A relative to B.

For the boundary plots in the next section, the following values (see Table 9) are used to define the decision model parameters. Each value is constant throughout a series of boundary plots, except for the parameter being investigated. For the function $c(t)$, the formulation determined from the CEC case study is also used. Additionally, a risk neutral designer is assumed.

Table 9. Decision Model Parameters for Boundary Plot Investigations

		Mean	Standard Deviatio
Product Prior	A ($M)	-5:5	0:1
	B ($M)	0	1.00
Analysis Parameters	Cost ($M)	0.1	-
	Quality ($M)	-	0.10
	Time (Years)	0.30	-
ZCV	($M)	-5	-

4.2.1 Analysis Monetary Cost

On the left side of each plot, *Select B* is the prescribed action, as the expected utility of B is larger than A. The opposite is true on the right side, where *Select A* is the prescribed action. In the middle, which corresponds to instances where the expected values of each alternative are similar, the designer should perform one or more analyses, depending on the standard deviation of alternative A. It is especially significant that the designer is never prescribed to perform an analysis on A instead of B. This is for two reasons. First it is assumed that the analyses have a common cost; if analyzing A was less expensive than B, then this may not be the case. Also, the standard deviation of A is strictly less than that of B due to normalization. Because the accuracies of the analyses on A and B are assumed to be equal, analyzing B will always reduce the uncertainty more than analyzing A. As such, the expected value of information for analyzing B will always exceed that of A.

Clearly, as the monetary cost of an analysis increases, its expected value of information should decrease. The Temporal Analysis Decision Model confirms this idea, as evidenced in the boundary plots of Figure 19.

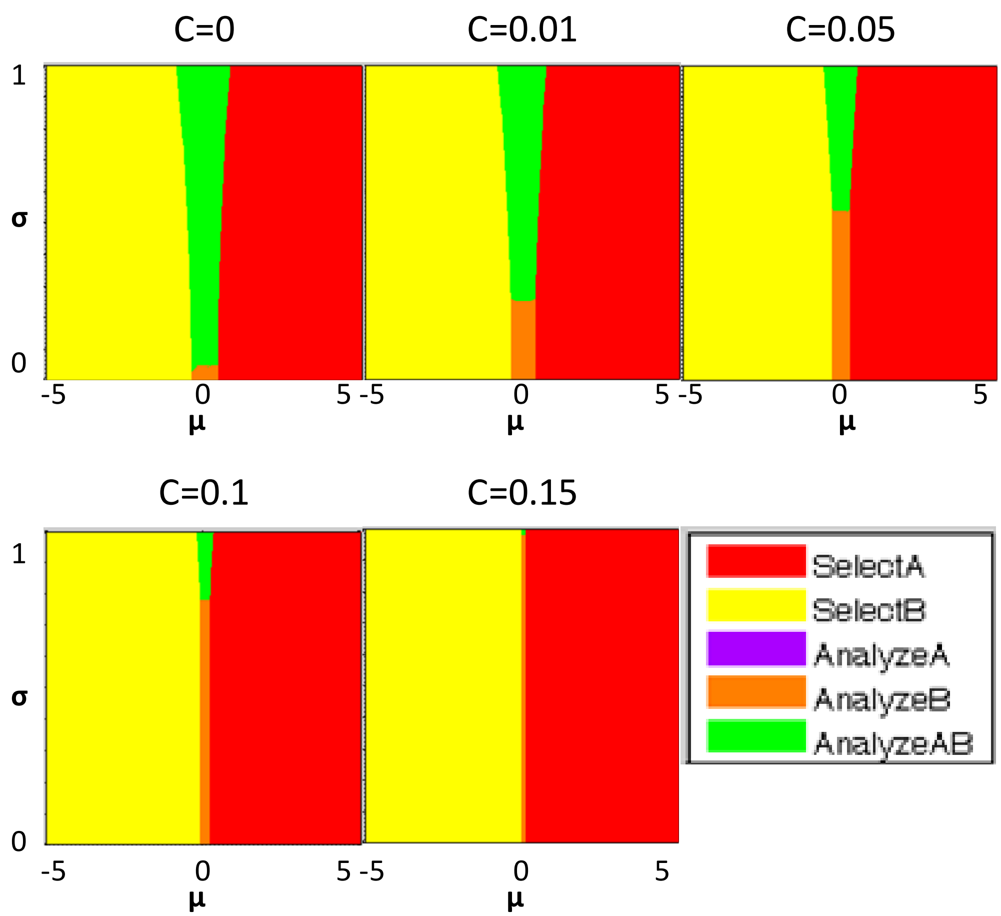


Figure 19. Boundary Plots of Varying Analysis Monetary Cost

As the cost of the analyses increase (Left to Right), the region in which it is the prescribed behavior to perform an analysis (either B or A&B) slowly reduces in size until the point is reached when analysis would never be prescribed. It is also noteworthy that even when the analyses have no monetary costs, they are still not recommended for all cases. This is due to the temporal costs of the analyses. It is also interesting that even when testing A is free, there appears to be scenarios where the designer should *Analyze B*, but not *Analyze A&B*. This is actually an error in the plotting of the regions; technically, the expected utilities of *Analyze B* and *Analyze A&B* are equivalent, and *Analyze B* is shown arbitrarily. They are equivalent because the uncertainty about A is already so small that performing an analysis on it does not noticeably reduce the uncertainty further. The effects of monetary cost on designer behavior are not of key interest in this thesis, so further comments are reserved for future work.

4.2.2 Analysis Accuracy

One would expect the accuracy of the analyses to also play a significant role in the determining which design action to prescribe. Figure 20 shows a series of boundary plots that vary in analysis accuracy so that this can be investigated.

Contrary to expectations, the boundary plots show that variations in analysis accuracy do not seem to play a major role in changing the prescriptive behavior of a designer. Even as analysis standard deviation approaches zero, indicating a perfect analysis, the Analysis regions do not grow in size noticeably. It appears that analyses must fulfill some necessary requirement of accuracy to be desirable, but that accuracy is not sufficient on its own to change prescribed behavior significantly beyond some level. However, as the analyses become less accurate, at some point the reductions in the expected value of information of the analyses begin to impact the prescribed behavior.

Eventually, this has the same effect of increasing cost, and analysis is never prescribed. Again, the effects of analysis quality on designer behavior are not of key interest in this thesis, so further comments are reserved for future work.

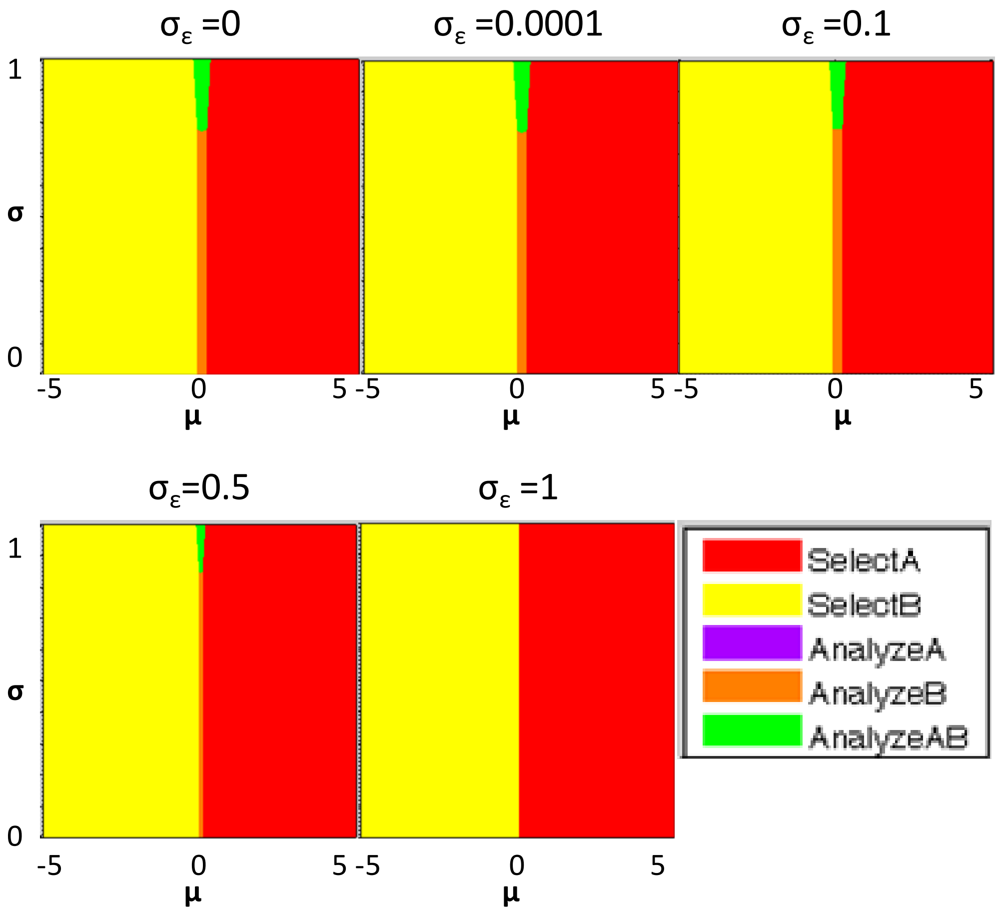


Figure 20. Boundary Plots of Varying Analysis Accuracy

4.2.3 Analysis Duration

In Chapter 1, two hypotheses were presented that proposed trends for prescribed designer behavior under the influence of temporally degrading product utility. Previously in this Chapter, two case studies were presented that provide empirical evidence to support these hypotheses. In this section, the hypotheses are further supported via the consideration of boundary plots that visualize the results of the Temporal Analysis Decision Model over varying parameter values.

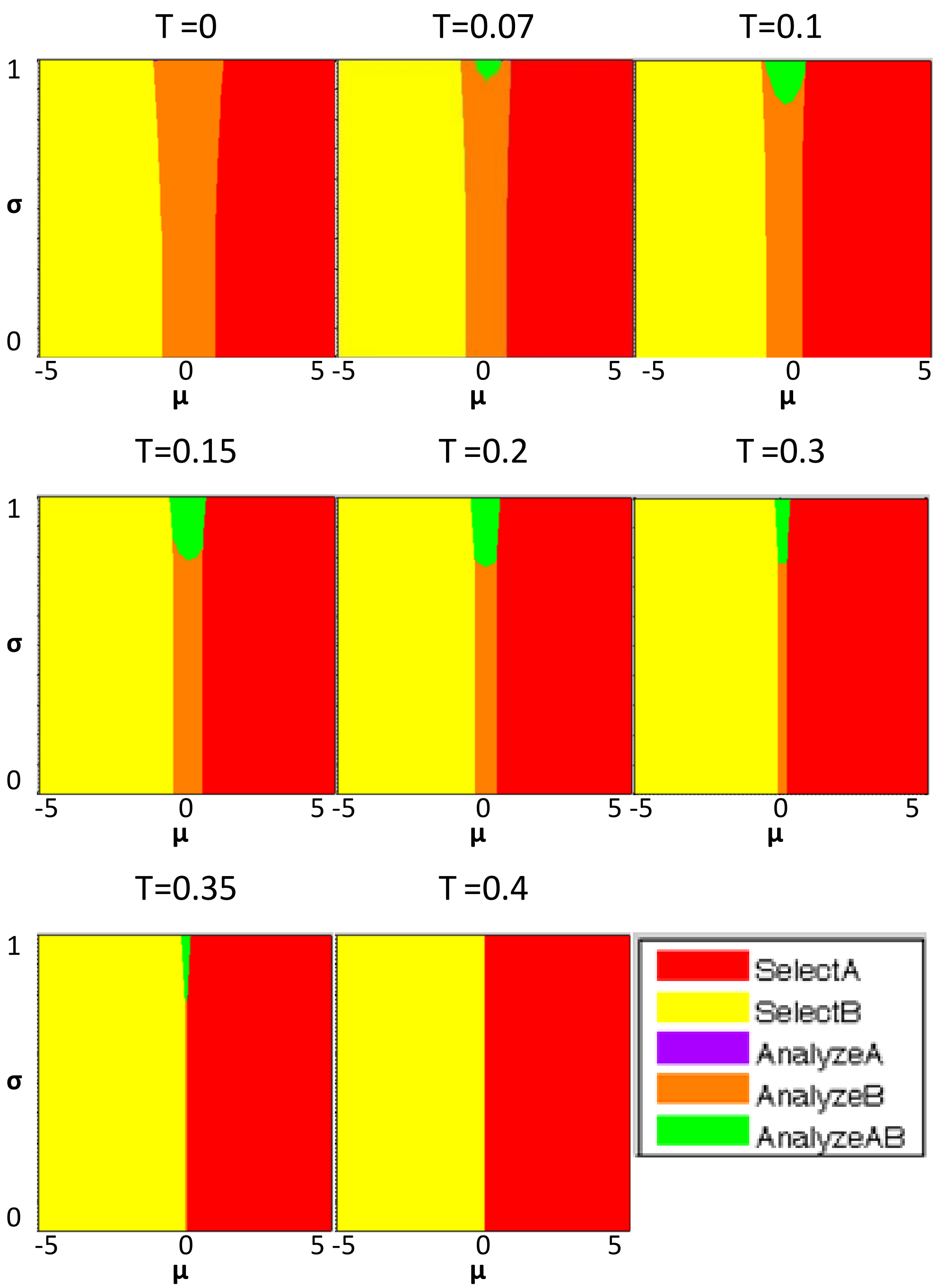


Figure 21. Boundary Plots of Varying Analysis Duration

In the boundary plot in which analysis duration is zero, an important concept is visualized. In the plot, the only options that are ever recommended are to *Analyze B* or to *Select A* or *B*. In other words, when analysis duration is not significant, designers should not perform analyses in parallel. Rather, any analyses performed should be done sequentially. Upon direct consideration of the Temporal Analysis Decision Model, this concept can be proven mathematically.

Consider a series of two analyses that can be performed: *Analyze A*, and *Analyze B*. For the sake of simplicity, assume that A will be analyzed first. For these series, the decision model describing the two possibilities of parallel and sequential is shown as a decision tree in Figure 22. Recall that in a decision tree, uncertain events are shown as circles with emanating arcs, and decisions are shown as boxes with emanating arcs.

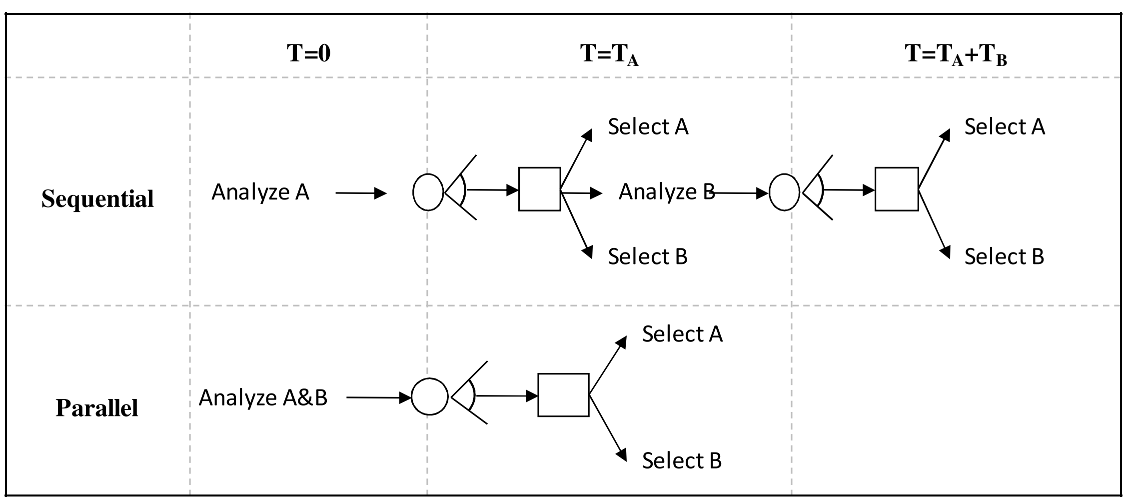


Figure 22. Comparison of Parallel and Sequential Analysis Strategies for Arbitrary Analysis Durations

In the figure, the actions are distinguished by time. The initial decision was already made at time T=0, but then the second decision is made after the analysis finishes, with a possible third decision being made after both analyses have completed sequentially. It has been shown that when time is valuable, this delay before selection of a final design can have an impact on the prescribed behavior for a designer. However, if $T_A = T_B = 0$, then degradation of product utility is not of concern. As a result, the decision tree can be visualized as shown in Figure 23.

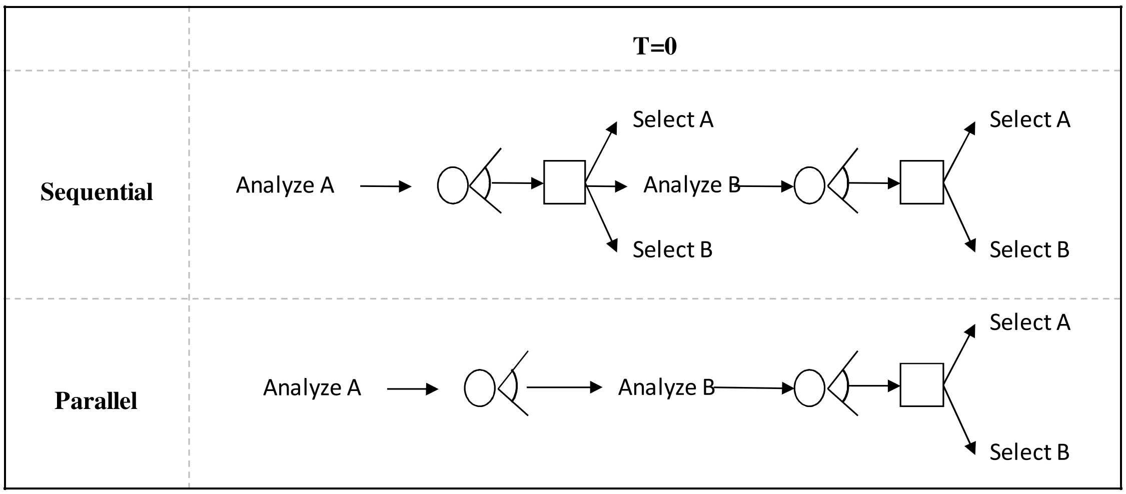


Figure 23. Comparison of Parallel and Sequential Analysis Strategies for Analysis Duration Equal to Zero

By visualizing the tree as such, the distinction between the two design strategies becomes clear: Sequential analysis allows the designer to utilize information from the first analysis before deciding whether to perform the second analysis. Since the decision model describes prescriptive behavior, at this additional decision node, the designer will always perform the action that maximizes the expected utility. Since the expected utility of the second analysis is identical for each strategy, and since the sequential strategy allows for two additional actions (*Select A* / *Select B*), the sequential strategy will always result in an expected utility that is greater than or equal that of parallel, so long as the duration of analysis is not significant.

This concept is important, as it supports the first hypothesis. According to the Temporal Analysis Decision Model, designers should not perform analyses in parallel if analysis duration is zero, or if product utility does not degrade with time. However, as discussed below, as product utility increasingly degrades due to analysis duration, the prescribed behavior of the designer increasingly trends towards parallel analysis.

Referring back to Figure 21, as the durations of analyses increases, or as the temporal costs of analyses increase, two phenomena are evidenced. The phenomena correspond directly to those proposed by the hypotheses.

The first phenomenon is the trend towards parallel testing. From the boundary plots, it is clear that as the duration of the analyses increase, a region of parallel testing appears and grows. This provides clear evidence to support the first hypothesis that the trend towards parallel analysis is a natural result of increasing temporal costs of design. It is noted that as the temporal costs continue to rise, the region of parallel testing begins to shrink, but this phenomenon only lends support to the second hypothesis.

The second phenomenon is trend towards risk acceptance through earlier selection of the design alternative. Notice that as the analysis duration increases, the thickness of the analysis regions begins to shrink. Eventually, the analyses become too “expensive” to perform, and the designer should simply select an alternative without any testing. This provides additional evidence to support the second hypothesis of this thesis.

Looking deeper into the parametric studies, it is not instantly visible why the trend in parallelization does not continue indefinitely. It is possible that the trend towards risk acceptance eventually dominates the trend of parallelization. But it is also possible that the trend itself is not monotonic, but rather peaks, then shifts directions. In order to investigate this further, the parametric study above was investigated further. The number of test cases for a particular analysis duration in which parallel analysis is the best decision was counted. This was then compared to the total number of test cases in which parallel or sequential analyses are the best decision. The ratio of parallel to total analysis test cases was computed and plotted for increasing analysis duration in Figure 24.

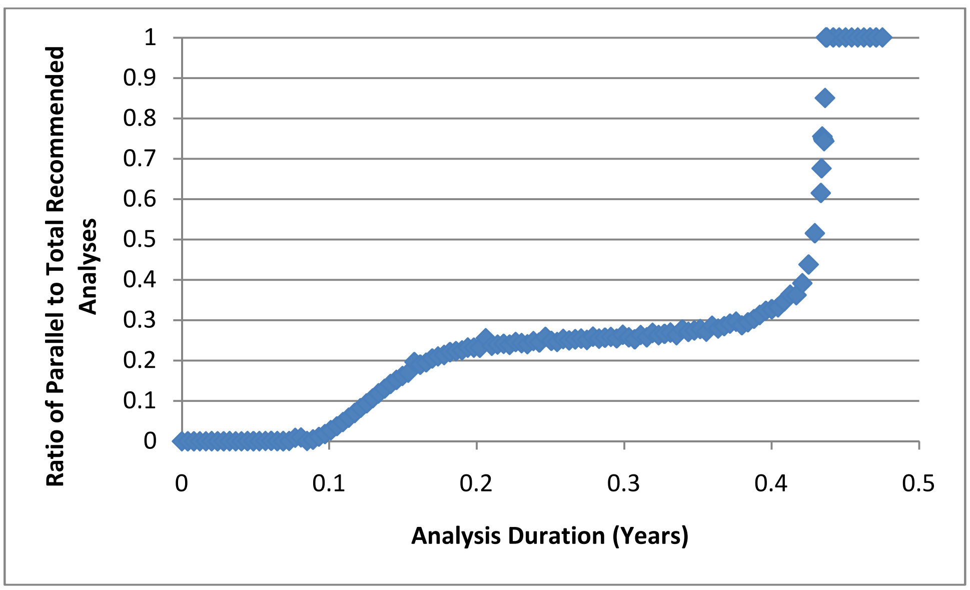


Figure 24. Ratio of Recommended Parallel Analyses to Total Recommended Analyses vs Analysis Duration

As shown in Figure 24, the ratio of recommend parallel analyses to total recommended analyses does appear to increase monotonically until it eventually reaches the case when all recommended analysis actions are parallel. It is noted that for sufficiently large analysis durations (ie. greater than 0.46 Years), no analysis is recommended. At this point, the trend towards risk acceptance has dominated the trend of parallelization, resulting in the recommendation to not perform any analysis. It is interesting that there appears to be two distinct sigmoidal features about t=0.15 Years and t=0.43 Years. It is not instantly clear why these sigmoids exist, but it is expected that they coincide with shifts in the c(t) coefficient function. Future work should investigate the ratio of parallel to total recommended analyses, as this may lead to additional conclusions about the underlying phenomena.

4.3 Summary

In this section, the Temporal Analysis Decision Model was leveraged to examine two case studies: an OEM parts supplier face with a deadline and a Consumer Electronics Company in a competitive marketplace. In both case studies, the prescribed behaviors for a designer were analyzed for scenarios in which product utility was either time-sensitive or not time-sensitive. Through comparisons of the behaviors as prescribed by the Temporal Analysis Decision Model, the case studies provided empirical evidence to support both hypotheses.

Also, boundary plots based upon evaluations of the Temporal Analysis Decision Model examined how variations in model parameters affected the prescribed behavior of designers. An important concept was extracted from the decision model in that a designer should never perform analyses in parallel if product utility is not time-sensitive. Then, the boundary plots were examined to provide additional evidence to support the hypotheses. Final statements about the validity of the hypotheses, as well as future work in the area are provided in the next Chapter.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Reviewing the Hypotheses

In Chapter 1, a problem was identified: The complexity of engineered systems is increasing, resulting in the increased complexity of designing these systems. Designers are faced with the consideration of multiple stakeholders, disciplines, and tradeoffs. However, these factors only relate to the specification of the final product. The designer must also consider how that specification is going to be defined and determine a way to manage all of these factors if a product is to be successful. It was argued that Utility Theory provides a simple metric for making decisions, and it was further argued in Chapter 2 that engineering design is an appropriate domain for the application of Utility Theory. This thesis has focused on the application of Utility Theory to investigate the research question introduced in Chapter 1.

How does the selection of design process activities depend on temporal degradation of product utility?

This research question is significant because its resolution will lead to new knowledge detailing how designers should behave when product utility is temporally sensitive. In Chapter 3, the literature was found to agree that it is very important to manage temporal costs of the design process in a proper manner. Recall that improper time management in design can lead to losses of profits, among other entities such as market share or product life.

The first hypothesis presented was,

H1: *When considering the temporal degradation of product utility, the maximization of expected utility leads to the parallelization of design tasks.*

In Chapter 4, two cases studies presented empirical validation of this hypothesis. For both case studies, the risk averse designer should decide to perform both analyses in parallel, rather than in sequence. Additionally, boundary plots were investigated and found to further support the hypothesis. It was specifically noted that when product utility does not degrade temporally, or if analysis duration is insignificantly small, that performing analyses in parallel is at best equally as favorable as in sequence. Therefore, it was declared that when product utility does not degrade with analysis duration, analyses should be performed in sequence, rather than in parallel. Since the boundary plots show the change in optimal decision towards parallelization of design tasks as time costs increased, the hypothesis is said to be validated.

The second hypothesis presented was,

H2: *When considering the temporal degradation of product utility, the maximization of expected utility leads to risk acceptance rather than risk mitigation*

In Chapter 4, the two case studies presented empirical validation of this hypothesis as well. For both case studies, the risk neutral designer should accept the uncertainty about the decision being faced, and not attempt to mitigate it through analysis. This trend is specific to the introduction of temporal degradation of product utility, as analysis is the prescribed behavior if product utility is not time sensitive. The boundary plots serve to support the hypothesis further, as it is clear that the region in which analysis is prescribed becomes smaller as temporal costs increase. It is notable that for sufficient temporal based costs, analysis should never be performed. Because of the evidence provided by the case studies and boundary plots, the hypothesis is declared to be validated.

5.2 Contributions

This work in this thesis has produced several contributions. In this section, they are enumerated and separated based on whether they are primarily of an academic nature or implementation-based.

Table 10. List of Contributions

Academic	Implementation
• Presented Temporal Analysis Decision Model.	• Implemented the Temporal Analysis Decision Model in MATLAB.
• Showed that increasing temporal costs initiate trend towards parallelization of design tasks.	• Implemented several simplifying assumptions into the TADM
• Showed that increasing temporal costs initiate trend towards risk acceptance.	• Developed and implemented a simple model describing the temporal degradation of product utility.
• Showed that designers should perform analyses in sequence if product utility does not degrade temporally.	• Evaluated the TADM in two case studies and determined prescribed behaviors for decision makers.
• Reviewed literature in the field of temporally conscious design.	• Evaluated the TADM in boundary plots used to visualize optimal

Academic	Implementation
	decisions for varying parameters.
• Defended engineering design as an appropriate application of Utility Theory.	
• Reviewed the methods of RD, RBD, and RID.	
• Structured the preference models of RD, RBD, and RID within Utility Theory and identified the limitations imposed by the methods.	

5.3 Limitations

This thesis investigated two trends in prescribed designer behavior under the particular circumstances of time-sensitive product utility. To make this investigation, a simple model for the manner in which value is affected by time was developed and applied to the Temporal Analysis Decision Model. The decision model was able to be solved due to the incorporation of several simplifying assumptions. These assumptions included normally distributed uncertainties of prior beliefs of product utilities and analyses. These assumptions appear reasonable in this work, but it is conceded that they are not likely to be exactly met in reality. As such, it would be beneficial to investigate if the trends would change under different set of beliefs as could be investigated using sampling based techniques.

Additionally, it is conceded that the Temporal Analysis Decision Model, or any similar decision model, is likely to be much too complicated to construct and evaluate for it to be useful for direct use in any real engineering scenario. The case studies and boundary plots presented in this thesis represent idealized scenarios, such that the model is simple to evaluate and analyze. It was therefore possible to leverage the decision model for this context. However, the model is not recommended for use external of that of an academic or research-based interest.

5.4 Future Work

A simple model of design was presented as Figure 1 in Chapter 1. According to the simple model, the design tasks of *Ideation*, *Analysis*, and *Evaluation* are iterated until a suitable design solution is *Selected*. In this thesis, however, the task of *Ideation* has been neglected and design alternatives were assumed to have already been defined. As such, the iterative nature of design has also been neglected. It is recognized that iteration is an important aspect of design, and future work in this field must be capable of considering it as well.

To that end, the next generation of the decision model will begin to more fully investigate the design process by addressing the design task of *Ideation*. Since *Ideation* is a creative process, it is difficult to model it in a structured manner. As a result, the initial design alternatives will remain unspecified at the top level of the decision model. However, the process of *Refinement* will be included as a form of *Ideation*. *Refinement* is a process in takes a design alternative concept and defines some aspect(s) such that the resulting design alternative is more descriptive. For example, assume a designer is making an automobile. AUTOMOBILE is the abstract design alternative concept. If the concept AUTOMOBILE is refined into two separate alternatives, say CAR and SUV, then CAR and SUV are the resulting design alternatives.

According to a preliminary model that has only been partially developed, a refined alternative has initial beliefs about value identical to that of the concept. It is only after the refined alternative has been analyzed that the beliefs are updated. The logic behind this model is most easily explained through the following series of figures.

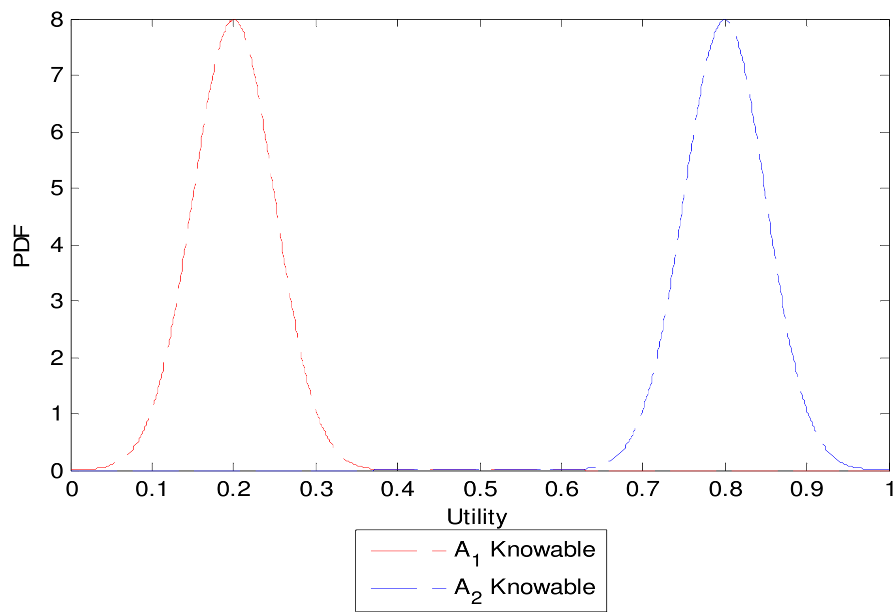


Figure 25. Knowable Utility for Two Refined Alternatives

In Figure 25, two probability distributions are shown that represent the possible utilities of two alternatives. Each pdf represents the state of knowledge about an alternative if the designer had perfect knowledge. It is noted that due to aleatory uncertainty, there will always be some variability, however this variability may be small. From the figure, it is clearly visible with perfect knowledge that Alternative A_2 (μ_2, σ_2^2) is vastly superior to Alternative A_1 (μ_1, σ_1^2).

However, it should be apparent that uncertainty can only be reduced to such levels if the alternatives have been analyzed. If they had not been analyzed, then there would be an additional amount of uncertainty that shall be termed σ_ε^2. Then, as shown in Figure 26, the expected values would stay the same, but the variances would increase to $\sigma_\varepsilon^2 + \sigma_1^2$ or $\sigma_\varepsilon^2 + \sigma_2^2$ for Alternatives A_1 and A_2, respectively.

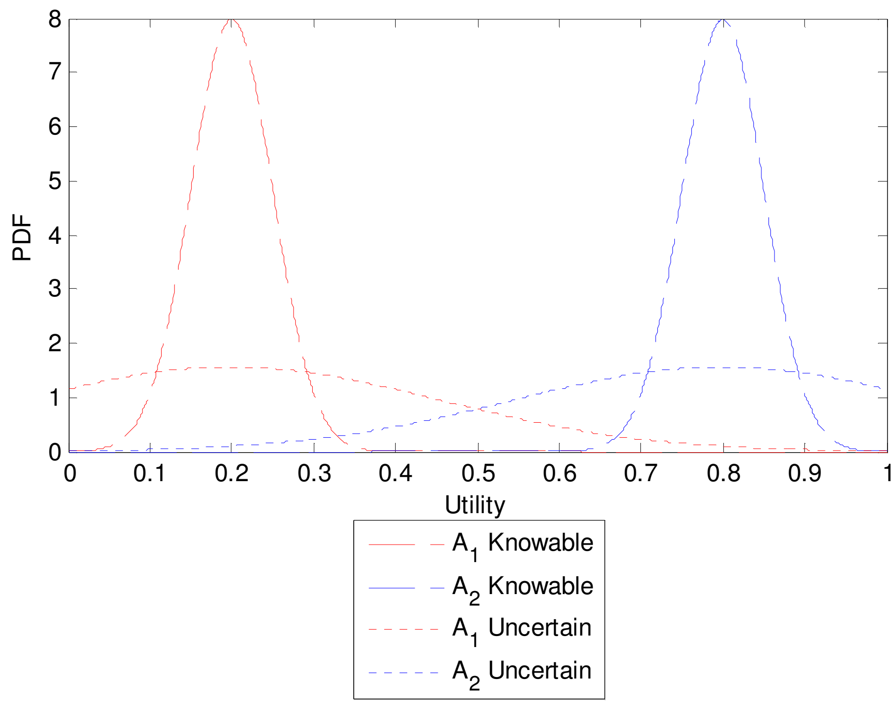
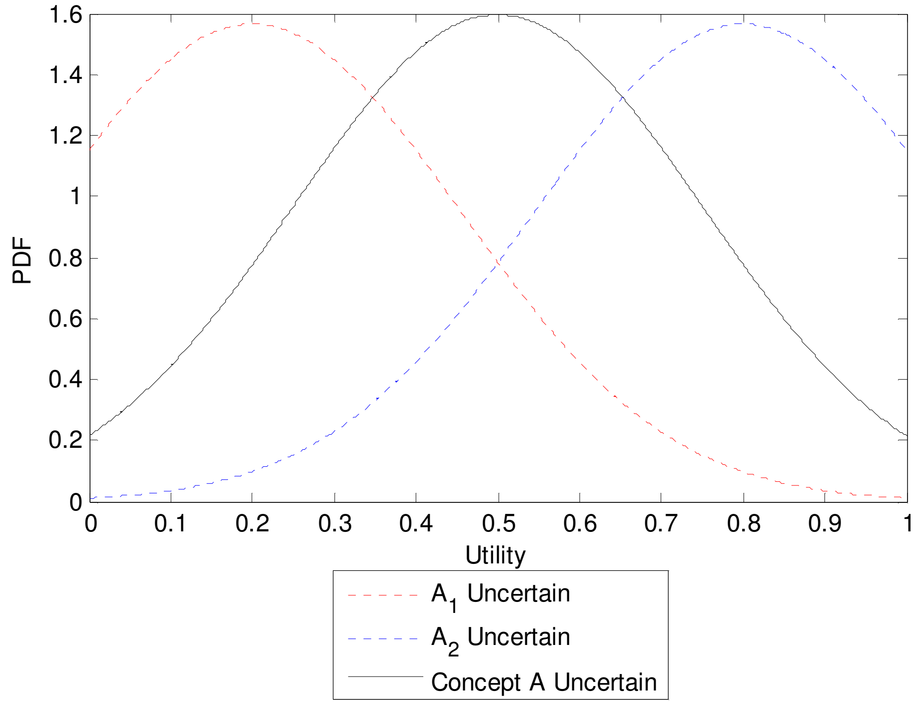


Figure 26. Knowable Utility Plus Uncertainty for Two Refined Alternatives

Then, if it is further assumed that it is uncertain which of the two Alternatives would be refined from Concept A first, then the maximization of entropy leads to the declaration of A as a sum of the two distributions A_1 and A_2, as shown in



.

Therefore, the mean of Concept A would be the average of the Alternatives. Further, if $\sigma_\varepsilon^2 \gg \sigma_1^2, \sigma_2^2$, then the uncertainty in a refined Alternative would be σ_ε^2, identical to the Concept. It would only be through analysis that Alternatives could be distinguished, and the decision model presented in this thesis already includes a model for analysis.

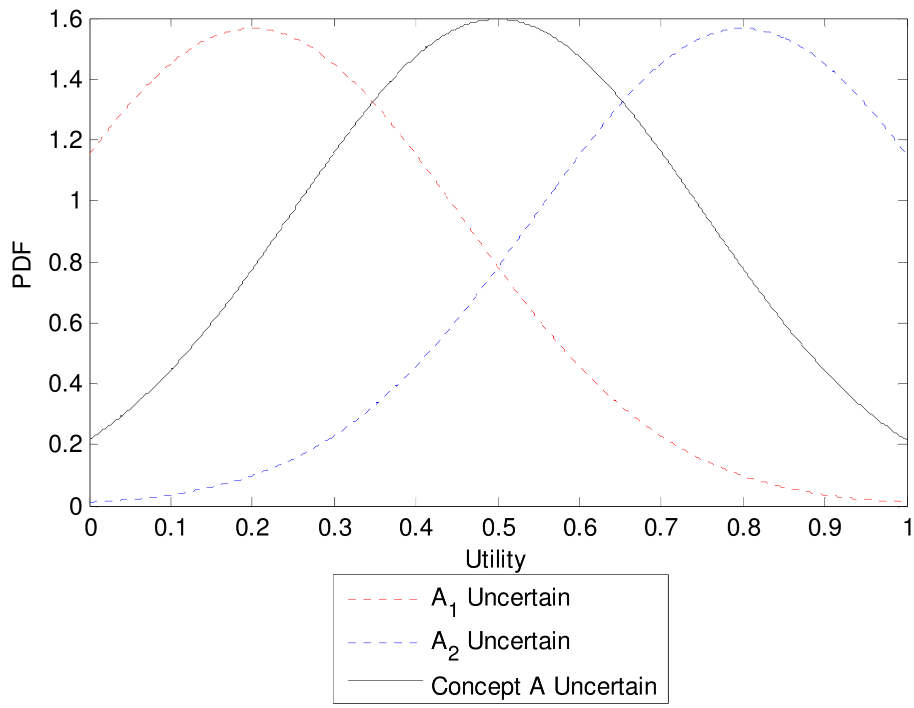


Figure 27. Utility for Concept and Refined Alternatives

The incorporation of the process of *Refinement* will transform the Temporal Analysis Decision Model into a Temporal Design Decision Model, and allow a much broader investigation of design process activities and trends. Investigations of the TDDM model will help design researchers to investigate phenomena such as the prescribed method to interchange between refinement and analysis to best reduce uncertainty in design.

APPENDIX

Relevant source code

SAE_Single_scenario1.m

```
% SAE_Single_scenario1.m is a script that calculates the expected
% utility of performing several different operations for the case
% studies.  The designer is faced with a decision to either select one
% of two design alternatives, or perform tests on them.  In this
% particular scenario, the tests take time to perform, and the
% values of the alternatives are affected by the current time. The
% options available to the designer initially are therefore:
% Select A
% Analyze A
% Analyze A&B at the same time
% Analyze B
% Select B
% If an analysis is performed, beliefs about the mean and standard
% deviation of the value of the alternative are updated using Bayesian
% statistics.  The designer must then decide then Select A or B or to
% perform further testing if they are still available.  The expected
% utility of each action is calculated for several different scenarios
% with the following as varying parameters.
% mu_vA_P - Prior Belief about the mean of the value of A
% mu_vB_P - Prior Belief about the mean of the value of B
% stdev_vA_P - Prior Belief about the standard deviation of the value
of A
% stdev_vB_P - Prior Belief about the standard deviation of the value
of B
% R - Risk coefficient
% C_A - The cost of Analyzing A
% C_B - The cost of Analyzing B
% stdev_epsA - Term describing the accuracy of the Analysis performed
on A
% stdev_epsB - Term describing the accuracy of the Analysis performed
on B
% T_A - The time required to perform the Analysis of A (0 = no time)
% T_B - The time required to perform the Analysis of B (1 = a long
time)

clear
clc
close all

addpath C:\Users\Ben\Documents\RiskAnalysis\RiskTime\Functions

%% Define Parameters
R=0;
C_A=.03;
C_B=.03;
stdev_epsA_P=.15;
```

```
stdev_epsB_P=.15;
T_A=2/3;      %in current format T_A<1,T_B<1,
T_B=2/3;
mu_vA_P=-1.5;
mu_vB_P=0;
stdev_vA_P=0.5;
stdev_vB_P=1;
tol=1e-6;
V_final=-4.5;

%Define value as a function of time
% val_time=V_init-(V_init-V_final)*c(t)
%   Because val_init is a normal distribution, the mean must be altered
by the coefficient c_time.

c_time=@(time) .5+.5*((time-1)/(eps+abs(time-1))); %step function, not
valid at t=1
% c_time=@(time) 0;
mu_time=@(v_init,time) v_init*(1-c_time(time))+V_final*c_time(time);
stdev_time=@(stdev_init,time) (1-c_time(time))*stdev_init;


%% R=0, Time Sensitive
%Define initial beliefs
infoState.mean=[mu_vA_P,mu_vB_P];
infoState.stdev=[stdev_vA_P,stdev_vB_P];
infoState.alternatives=[1,2];

%Expectation of selecting immediately A/B
mu_vA_t0=mu_time(mu_vA_P,0);
mu_vB_t0=mu_time(mu_vB_P,0);
stdev_vA_t0=stdev_time(stdev_vA_P,0);
stdev_vB_t0=stdev_time(stdev_vB_P,0);
E_SelectA=utilFunction(mu_vA_t0,stdev_vA_t0,R);
E_SelectB=utilFunction(mu_vB_t0,stdev_vB_t0,R);

%Expectation of testing A
mu_vA_tA1=mu_time(mu_vA_t0,T_A);
mu_vB_tA=mu_time(mu_vB_t0,T_A);
stdev_vA_tA1=stdev_time(stdev_vA_t0,T_A);
stdev_vB_tA=stdev_time(stdev_vB_t0,T_A);
stdev_epsA=stdev_time(stdev_epsA_P,0);
mu_vA_tA=@(mu_testA)
(mu_vA_tA1*stdev_epsA^2+mu_testA*stdev_vA_tA1^2)/(stdev_epsA^2+stdev_vA
_tA1^2);
stdev_vA_tA=sqrt((stdev_epsA^2*stdev_vA_tA1^2)/(stdev_epsA^2+stdev_vA_t
A1^2));

mu_vA_tAB=@(mu_testA) mu_time((mu_vA_tA(mu_testA)-
c_time(T_A)*V_final)/(1-c_time(T_A)),T_A+T_B);
mu_vB_tAB=mu_time(mu_vB_t0,T_A+T_B);
stdev_vA_tAB=stdev_time(stdev_vA_tA/(1-c_time(T_A)),T_A+T_B);
stdev_vB_tAB=stdev_time(stdev_vB_t0,T_A+T_B);
stdev_epsB=stdev_time(stdev_epsB_P,T_A);
%integral over mu_testA
```

```
AA_AB=@(mu_testA)
E_U_Test_X(mu_vA_tAB(mu_testA),mu_vB_tAB,stdev_vA_tAB,stdev_vB_tAB,C_A+
C_B,stdev_epsB,R,2);
AA_SA=@(mu_testA) utilFunction(mu_vA_tA(mu_testA)-C_A,stdev_vA_tA,R);
AA_SB=utilFunction(mu_vB_tA-C_A,stdev_vB_tA,R);
E_AnalyzeA=quadv(@(mu_testA)
max(max(AA_SA(mu_testA),AA_SB),AA_AB(mu_testA)).*normpdf(mu_testA,mu_vA
_tA1,sqrt(stdev_vA_tA1^2+stdev_epsA^2)),mu_vA_tA1-
10*stdev_vA_tA1,mu_vA_tA1+10*stdev_vA_tA1,tol);

%Expectation of testing B
mu_vB_tB1=mu_time(mu_vB_t0,T_B);
mu_vA_tB=mu_time(mu_vA_t0,T_B);
stdev_vB_tB1=stdev_time(stdev_vB_t0,T_B);
stdev_vA_tB=stdev_time(stdev_vA_t0,T_B);
stdev_epsB=stdev_time(stdev_epsB_P,0);
mu_vB_tB=@(mu_testB)
(mu_vB_tB1*stdev_epsB^2+mu_testB*stdev_vB_tB1^2)/(stdev_epsB^2+stdev_vB
_tB1^2);
stdev_vB_tB=sqrt((stdev_epsB^2*stdev_vB_tB1^2)/(stdev_epsB^2+stdev_vB_t
B1^2));

mu_vB_tBA=@(mu_testB) mu_time((mu_vB_tB(mu_testB)-
c_time(T_B)*V_final)/(1-c_time(T_B)),T_A+T_B);
mu_vA_tBA=mu_time(mu_vA_t0,T_A+T_B);
stdev_vB_tBA=stdev_time(stdev_vB_tB/(1-c_time(T_B)),T_A+T_B);
stdev_vA_tBA=stdev_time(stdev_vA_t0,T_A+T_B);
stdev_epsA=stdev_time(stdev_epsA_P,T_B);
%integral over mu_testB

AB_AA=@(mu_testB)
E_U_Test_X(mu_vA_tBA,mu_vB_tBA(mu_testB),stdev_vA_tBA,stdev_vB_tBA,C_A+
C_B,stdev_epsA,R,1);
AB_SB=@(mu_testB) utilFunction(mu_vB_tB(mu_testB)-C_B,stdev_vB_tB,R);
AB_SA=utilFunction(mu_vA_tB-C_B,stdev_vA_tB,R);
E_AnalyzeB=quadv(@(mu_testB)
max(max(AB_SB(mu_testB),AB_SA),AB_AA(mu_testB)).*normpdf(mu_testB,mu_vB
_tB1,sqrt(stdev_vB_tB1^2+stdev_epsB^2)),mu_vB_tB1-
10*stdev_vB_tB1,mu_vB_tB1+10*stdev_vB_tB1,tol);


%Expectation of testing A and B
if T_A<=T_B
 mu_vA_tA1=mu_time(mu_vA_t0,T_A);
mu_vB_tA=mu_time(mu_vB_t0,T_A);
stdev_vA_tA1=stdev_time(stdev_vA_t0,T_A);
stdev_vB_tA=stdev_time(stdev_vB_t0,T_A);
stdev_epsA=stdev_time(stdev_epsA,0);
stdev_epsB=stdev_time(stdev_epsB,0);
mu_vA_tA=@(mu_testA)
(mu_vA_tA1*stdev_epsA^2+mu_testA*stdev_vA_tA1^2)/(stdev_epsA^2+stdev_vA
_tA1^2);
stdev_vA_tA=sqrt((stdev_epsA^2*stdev_vA_tA1^2)/(stdev_epsA^2+stdev_vA_t
A1^2));
```

```
mu_vA_tAB=@(mu_testA) mu_time((mu_vA_tA(mu_testA)-
c_time(T_A)*V_final)/(1-c_time(T_A)),T_B);
mu_vB_tAB=mu_time(mu_vB_t0,T_B);
stdev_vA_tAB=stdev_time(stdev_vA_tA/(1-c_time(T_A)),T_B);
stdev_vB_tAB=stdev_time(stdev_vB_t0,T_B);
%integral over mu_testA

AAB=@(mu_testA)
E_U_Test_X(mu_vA_tAB(mu_testA),mu_vB_tAB,stdev_vA_tAB,stdev_vB_tAB,C_A+
C_B,stdev_epsB,R,2);
E_AnalyzeAB=quadv(@(mu_testA)
AAB(mu_testA).*normpdf(mu_testA,mu_vA_tA1,sqrt(stdev_vA_tA1^2+stdev_eps
A^2)),mu_vA_tA1-10*stdev_vA_tA1,mu_vA_tA1+10*stdev_vA_tA1,tol);

else %T_A>T_B
mu_vB_tB1=mu_time(mu_vB_t0,T_B);
mu_vA_tB=mu_time(mu_vA_t0,T_B);
stdev_vB_tB1=stdev_time(stdev_vB_t0,T_B);
stdev_vA_tB=stdev_time(stdev_vA_t0,T_B);
stdev_epsA=stdev_time(stdev_epsA,0);
stdev_epsB=stdev_time(stdev_epsB,0);
mu_vB_tB=@(mu_testB)
(mu_vB_tB1*stdev_epsB^2+mu_testB*stdev_vB_tB1^2)/(stdev_epsB^2+stdev_vB
_tB1^2);
stdev_vB_tB=sqrt((stdev_epsB^2*stdev_vB_tB1^2)/(stdev_epsB^2+stdev_vB_t
B1^2));

mu_vB_tBA=@(mu_testB) mu_time((mu_vB_tB(mu_testB)-
c_time(T_B)*V_final)/(1-c_time(T_B)),T_A);
mu_vA_tBA=mu_time(mu_vA_t0,T_A);
stdev_vB_tBA=stdev_time(stdev_vB_tB/(1-c_time(T_B)),T_A);
stdev_vA_tBA=stdev_time(stdev_vA_t0,T_A);
%integral over mu_testB

ABA=@(mu_testB)
E_U_Test_X(mu_vA_tBA,mu_vB_tBA(mu_testB),stdev_vA_tBA,stdev_vB_tBA,C_A+
C_B,stdev_epsA,R,1);
E_AnalyzeAB=quadv(@(mu_testB)
ABA(mu_testB).*normpdf(mu_testB,mu_vB_tB1,sqrt(stdev_vB_tB1^2+stdev_eps
B^2)),mu_vB_tB1-10*stdev_vB_tB1,mu_vB_tB1+10*stdev_vB_tB1,tol);
end

dec_R0_ct=[E_SelectA;E_SelectB;E_AnalyzeA;E_AnalyzeB;E_AnalyzeAB]

%% R=1, Time Sensitive
R=1;
infoState.mean=[mu_vA_P,mu_vB_P];
infoState.stdev=[stdev_vA_P,stdev_vB_P];
infoState.alternatives=[1,2];

%Expectation of selecting immediately A/B
mu_vA_t0=mu_time(mu_vA_P,0);
mu_vB_t0=mu_time(mu_vB_P,0);
stdev_vA_t0=stdev_time(stdev_vA_P,0);
```

```
stdev_vB_t0=stdev_time(stdev_vB_P,0);
E_SelectA=utilFunction(mu_vA_t0,stdev_vA_t0,R);
E_SelectB=utilFunction(mu_vB_t0,stdev_vB_t0,R);

%Expectation of testing A
mu_vA_tA1=mu_time(mu_vA_t0,T_A);
mu_vB_tA=mu_time(mu_vB_t0,T_A);
stdev_vA_tA1=stdev_time(stdev_vA_t0,T_A);
stdev_vB_tA=stdev_time(stdev_vB_t0,T_A);
stdev_epsA=stdev_time(stdev_epsA_P,0);
mu_vA_tA=@(mu_testA)
(mu_vA_tA1*stdev_epsA^2+mu_testA*stdev_vA_tA1^2)/(stdev_epsA^2+stdev_vA
_tA1^2);
stdev_vA_tA=sqrt((stdev_epsA^2*stdev_vA_tA1^2)/(stdev_epsA^2+stdev_vA_t
A1^2));

mu_vA_tAB=@(mu_testA) mu_time((mu_vA_tA(mu_testA)-
c_time(T_A)*V_final)/(1-c_time(T_A)),T_A+T_B);
mu_vB_tAB=mu_time(mu_vB_t0,T_A+T_B);
stdev_vA_tAB=stdev_time(stdev_vA_tA/(1-c_time(T_A)),T_A+T_B);
stdev_vB_tAB=stdev_time(stdev_vB_t0,T_A+T_B);
stdev_epsB=stdev_time(stdev_epsB_P,T_A);
%integral over mu_testA

AA_AB=@(mu_testA)
E_U_Test_X(mu_vA_tAB(mu_testA),mu_vB_tAB,stdev_vA_tAB,stdev_vB_tAB,C_A+
C_B,stdev_epsB,R,2);
AA_SA=@(mu_testA) utilFunction(mu_vA_tA(mu_testA)-C_A,stdev_vA_tA,R);
AA_SB=utilFunction(mu_vB_tA-C_A,stdev_vB_tA,R);
E_AnalyzeA=quadv(@(mu_testA)
max(max(AA_SA(mu_testA),AA_SB),AA_AB(mu_testA)).*normpdf(mu_testA,mu_vA
_tA1,sqrt(stdev_vA_tA1^2+stdev_epsA^2)),mu_vA_tA1-
10*stdev_vA_tA1,mu_vA_tA1+10*stdev_vA_tA1,tol);

%Expectation of testing B
mu_vB_tB1=mu_time(mu_vB_t0,T_B);
mu_vA_tB=mu_time(mu_vA_t0,T_B);
stdev_vB_tB1=stdev_time(stdev_vB_t0,T_B);
stdev_vA_tB=stdev_time(stdev_vA_t0,T_B);
stdev_epsB=stdev_time(stdev_epsB_P,0);
mu_vB_tB=@(mu_testB)
(mu_vB_tB1*stdev_epsB^2+mu_testB*stdev_vB_tB1^2)/(stdev_epsB^2+stdev_vB
_tB1^2);
stdev_vB_tB=sqrt((stdev_epsB^2*stdev_vB_tB1^2)/(stdev_epsB^2+stdev_vB_t
B1^2));

mu_vB_tBA=@(mu_testB) mu_time((mu_vB_tB(mu_testB)-
c_time(T_B)*V_final)/(1-c_time(T_B)),T_A+T_B);
mu_vA_tBA=mu_time(mu_vA_t0,T_A+T_B);
stdev_vB_tBA=stdev_time(stdev_vB_tB/(1-c_time(T_B)),T_A+T_B);
stdev_vA_tBA=stdev_time(stdev_vA_t0,T_A+T_B);
stdev_epsA=stdev_time(stdev_epsA_P,T_B);
%integral over mu_testB
```

```
AB_AA=@(mu_testB)
E_U_Test_X(mu_vA_tBA,mu_vB_tBA(mu_testB),stdev_vA_tBA,stdev_vB_tBA,C_A+
C_B,stdev_epsA,R,1);
AB_SB=@(mu_testB) utilFunction(mu_vB_tB(mu_testB)-C_B,stdev_vB_tB,R);
AB_SA=utilFunction(mu_vA_tB-C_B,stdev_vA_tB,R);
E_AnalyzeB=quadv(@(mu_testB)
max(max(AB_SB(mu_testB),AB_SA),AB_AA(mu_testB)).*normpdf(mu_testB,mu_vB
_tB1,sqrt(stdev_vB_tB1^2+stdev_epsB^2)),mu_vB_tB1-
10*stdev_vB_tB1,mu_vB_tB1+10*stdev_vB_tB1,tol);


%Expectation of testing A and B
if T_A<=T_B
 mu_vA_tA1=mu_time(mu_vA_t0,T_A);
mu_vB_tA=mu_time(mu_vB_t0,T_A);
stdev_vA_tA1=stdev_time(stdev_vA_t0,T_A);
stdev_vB_tA=stdev_time(stdev_vB_t0,T_A);
stdev_epsA=stdev_time(stdev_epsA,0);
stdev_epsB=stdev_time(stdev_epsB,0);
mu_vA_tA=@(mu_testA)
(mu_vA_tA1*stdev_epsA^2+mu_testA*stdev_vA_tA1^2)/(stdev_epsA^2+stdev_vA
_tA1^2);
stdev_vA_tA=sqrt((stdev_epsA^2*stdev_vA_tA1^2)/(stdev_epsA^2+stdev_vA_t
A1^2));

mu_vA_tAB=@(mu_testA) mu_time((mu_vA_tA(mu_testA)-
c_time(T_A)*V_final)/(1-c_time(T_A)),T_B);
mu_vB_tAB=mu_time(mu_vB_t0,T_B);
stdev_vA_tAB=stdev_time(stdev_vA_tA/(1-c_time(T_A)),T_B);
stdev_vB_tAB=stdev_time(stdev_vB_t0,T_B);
%integral over mu_testA

AAB=@(mu_testA)
E_U_Test_X(mu_vA_tAB(mu_testA),mu_vB_tAB,stdev_vA_tAB,stdev_vB_tAB,C_A+
C_B,stdev_epsB,R,2);
E_AnalyzeAB=quadv(@(mu_testA)
AAB(mu_testA).*normpdf(mu_testA,mu_vA_tA1,sqrt(stdev_vA_tA1^2+stdev_eps
A^2)),mu_vA_tA1-10*stdev_vA_tA1,mu_vA_tA1+10*stdev_vA_tA1,tol);

else %T_A>T_B
mu_vB_tB1=mu_time(mu_vB_t0,T_B);
mu_vA_tB=mu_time(mu_vA_t0,T_B);
stdev_vB_tB1=stdev_time(stdev_vB_t0,T_B);
stdev_vA_tB=stdev_time(stdev_vA_t0,T_B);
stdev_epsA=stdev_time(stdev_epsA,0);
stdev_epsB=stdev_time(stdev_epsB,0);
mu_vB_tB=@(mu_testB)
(mu_vB_tB1*stdev_epsB^2+mu_testB*stdev_vB_tB1^2)/(stdev_epsB^2+stdev_vB
_tB1^2);
stdev_vB_tB=sqrt((stdev_epsB^2*stdev_vB_tB1^2)/(stdev_epsB^2+stdev_vB_t
B1^2));

mu_vB_tBA=@(mu_testB) mu_time((mu_vB_tB(mu_testB)-
c_time(T_B)*V_final)/(1-c_time(T_B)),T_A);
mu_vA_tBA=mu_time(mu_vA_t0,T_A);
stdev_vB_tBA=stdev_time(stdev_vB_tB/(1-c_time(T_B)),T_A);
```

```
stdev_vA_tBA=stdev_time(stdev_vA_t0,T_A);
%integral over mu_testB

ABA=@(mu_testB)
E_U_Test_X(mu_vA_tBA,mu_vB_tBA(mu_testB),stdev_vA_tBA,stdev_vB_tBA,C_A+
C_B,stdev_epsA,R,1);
E_AnalyzeAB=quadv(@(mu_testB)
ABA(mu_testB).*normpdf(mu_testB,mu_vB_tB1,sqrt(stdev_vB_tB1^2+stdev_eps
B^2)),mu_vB_tB1-10*stdev_vB_tB1,mu_vB_tB1+10*stdev_vB_tB1,tol);
end

dec_R1_ct=[E_SelectA;E_SelectB;E_AnalyzeA;E_AnalyzeB;E_AnalyzeAB]

%% R=0, Not Time Sensitive
c_time=@(time) 0;
mu_time=@(v_init,time) v_init*(1-c_time(time))+V_final*c_time(time);
stdev_time=@(stdev_init,time) (1-c_time(time))*stdev_init;
R=0;

infoState.mean=[mu_vA_P,mu_vB_P];
infoState.stdev=[stdev_vA_P,stdev_vB_P];
infoState.alternatives=[1,2];

%Expectation of selecting immediately A/B
mu_vA_t0=mu_time(mu_vA_P,0);
mu_vB_t0=mu_time(mu_vB_P,0);
stdev_vA_t0=stdev_time(stdev_vA_P,0);
stdev_vB_t0=stdev_time(stdev_vB_P,0);
E_SelectA=utilFunction(mu_vA_t0,stdev_vA_t0,R);
E_SelectB=utilFunction(mu_vB_t0,stdev_vB_t0,R);

%Expectation of testing A
mu_vA_tA1=mu_time(mu_vA_t0,T_A);
mu_vB_tA=mu_time(mu_vB_t0,T_A);
stdev_vA_tA1=stdev_time(stdev_vA_t0,T_A);
stdev_vB_tA=stdev_time(stdev_vB_t0,T_A);
stdev_epsA=stdev_time(stdev_epsA_P,0);
mu_vA_tA=@(mu_testA)
(mu_vA_tA1*stdev_epsA^2+mu_testA*stdev_vA_tA1^2)/(stdev_epsA^2+stdev_vA
_tA1^2);
stdev_vA_tA=sqrt((stdev_epsA^2*stdev_vA_tA1^2)/(stdev_epsA^2+stdev_vA_t
A1^2));

mu_vA_tAB=@(mu_testA) mu_time((mu_vA_tA(mu_testA)-
c_time(T_A)*V_final)/(1-c_time(T_A)),T_A+T_B);
mu_vB_tAB=mu_time(mu_vB_t0,T_A+T_B);
stdev_vA_tAB=stdev_time(stdev_vA_tA/(1-c_time(T_A)),T_A+T_B);
stdev_vB_tAB=stdev_time(stdev_vB_t0,T_A+T_B);
stdev_epsB=stdev_time(stdev_epsB_P,T_A);
%integral over mu_testA

AA_AB=@(mu_testA)
E_U_Test_X(mu_vA_tAB(mu_testA),mu_vB_tAB,stdev_vA_tAB,stdev_vB_tAB,C_A+
C_B,stdev_epsB,R,2);
AA_SA=@(mu_testA) utilFunction(mu_vA_tA(mu_testA)-C_A,stdev_vA_tA,R);
```

```
AA_SB=utilFunction(mu_vB_tA-C_A,stdev_vB_tA,R);
E_AnalyzeA=quadv(@(mu_testA)
max(max(AA_SA(mu_testA),AA_SB),AA_AB(mu_testA)).*normpdf(mu_testA,mu_vA
_tA1,sqrt(stdev_vA_tA1^2+stdev_epsA^2)),mu_vA_tA1-
10*stdev_vA_tA1,mu_vA_tA1+10*stdev_vA_tA1,tol);

%Expectation of testing B
mu_vB_tB1=mu_time(mu_vB_t0,T_B);
mu_vA_tB=mu_time(mu_vA_t0,T_B);
stdev_vB_tB1=stdev_time(stdev_vB_t0,T_B);
stdev_vA_tB=stdev_time(stdev_vA_t0,T_B);
stdev_epsB=stdev_time(stdev_epsB_P,0);
mu_vB_tB=@(mu_testB)
(mu_vB_tB1*stdev_epsB^2+mu_testB*stdev_vB_tB1^2)/(stdev_epsB^2+stdev_vB
_tB1^2);
stdev_vB_tB=sqrt((stdev_epsB^2*stdev_vB_tB1^2)/(stdev_epsB^2+stdev_vB_t
B1^2));

mu_vB_tBA=@(mu_testB) mu_time((mu_vB_tB(mu_testB)-
c_time(T_B)*V_final)/(1-c_time(T_B)),T_A+T_B);
mu_vA_tBA=mu_time(mu_vA_t0,T_A+T_B);
stdev_vB_tBA=stdev_time(stdev_vB_tB/(1-c_time(T_B)),T_A+T_B);
stdev_vA_tBA=stdev_time(stdev_vA_t0,T_A+T_B);
stdev_epsA=stdev_time(stdev_epsA_P,T_B);
%integral over mu_testB

AB_AA=@(mu_testB)
E_U_Test_X(mu_vA_tBA,mu_vB_tBA(mu_testB),stdev_vA_tBA,stdev_vB_tBA,C_A+
C_B,stdev_epsA,R,1);
AB_SB=@(mu_testB) utilFunction(mu_vB_tB(mu_testB)-C_B,stdev_vB_tB,R);
AB_SA=utilFunction(mu_vA_tB-C_B,stdev_vA_tB,R);
E_AnalyzeB=quadv(@(mu_testB)
max(max(AB_SB(mu_testB),AB_SA),AB_AA(mu_testB)).*normpdf(mu_testB,mu_vB
_tB1,sqrt(stdev_vB_tB1^2+stdev_epsB^2)),mu_vB_tB1-
10*stdev_vB_tB1,mu_vB_tB1+10*stdev_vB_tB1,tol);


%Expectation of testing A and B
if T_A<=T_B
 mu_vA_tA1=mu_time(mu_vA_t0,T_A);
mu_vB_tA=mu_time(mu_vB_t0,T_A);
stdev_vA_tA1=stdev_time(stdev_vA_t0,T_A);
stdev_vB_tA=stdev_time(stdev_vB_t0,T_A);
stdev_epsA=stdev_time(stdev_epsA,0);
stdev_epsB=stdev_time(stdev_epsB,0);
mu_vA_tA=@(mu_testA)
(mu_vA_tA1*stdev_epsA^2+mu_testA*stdev_vA_tA1^2)/(stdev_epsA^2+stdev_vA
_tA1^2);
stdev_vA_tA=sqrt((stdev_epsA^2*stdev_vA_tA1^2)/(stdev_epsA^2+stdev_vA_t
A1^2));

mu_vA_tAB=@(mu_testA) mu_time((mu_vA_tA(mu_testA)-
c_time(T_A)*V_final)/(1-c_time(T_A)),T_B);
mu_vB_tAB=mu_time(mu_vB_t0,T_B);
stdev_vA_tAB=stdev_time(stdev_vA_tA/(1-c_time(T_A)),T_B);
stdev_vB_tAB=stdev_time(stdev_vB_t0,T_B);
```

```
%integral over mu_testA

AAB=@(mu_testA)
E_U_Test_X(mu_vA_tAB(mu_testA),mu_vB_tAB,stdev_vA_tAB,stdev_vB_tAB,C_A+
C_B,stdev_epsB,R,2);
E_AnalyzeAB=quadv(@(mu_testA)
AAB(mu_testA).*normpdf(mu_testA,mu_vA_tA1,sqrt(stdev_vA_tA1^2+stdev_eps
A^2)),mu_vA_tA1-10*stdev_vA_tA1,mu_vA_tA1+10*stdev_vA_tA1,tol);

else %T_A>T_B
mu_vB_tB1=mu_time(mu_vB_t0,T_B);
mu_vA_tB=mu_time(mu_vA_t0,T_B);
stdev_vB_tB1=stdev_time(stdev_vB_t0,T_B);
stdev_vA_tB=stdev_time(stdev_vA_t0,T_B);
stdev_epsA=stdev_time(stdev_epsA,0);
stdev_epsB=stdev_time(stdev_epsB,0);
mu_vB_tB=@(mu_testB)
(mu_vB_tB1*stdev_epsB^2+mu_testB*stdev_vB_tB1^2)/(stdev_epsB^2+stdev_vB
_tB1^2);
stdev_vB_tB=sqrt((stdev_epsB^2*stdev_vB_tB1^2)/(stdev_epsB^2+stdev_vB_t
B1^2));

mu_vB_tBA=@(mu_testB) mu_time((mu_vB_tB(mu_testB)-
c_time(T_B)*V_final)/(1-c_time(T_B)),T_A);
mu_vA_tBA=mu_time(mu_vA_t0,T_A);
stdev_vB_tBA=stdev_time(stdev_vB_tB/(1-c_time(T_B)),T_A);
stdev_vA_tBA=stdev_time(stdev_vA_t0,T_A);
%integral over mu_testB

ABA=@(mu_testB)
E_U_Test_X(mu_vA_tBA,mu_vB_tBA(mu_testB),stdev_vA_tBA,stdev_vB_tBA,C_A+
C_B,stdev_epsA,R,1);
E_AnalyzeAB=quadv(@(mu_testB)
ABA(mu_testB).*normpdf(mu_testB,mu_vB_tB1,sqrt(stdev_vB_tB1^2+stdev_eps
B^2)),mu_vB_tB1-10*stdev_vB_tB1,mu_vB_tB1+10*stdev_vB_tB1,tol);
end

dec_R0_c0=[E_SelectA;E_SelectB;E_AnalyzeA;E_AnalyzeB;E_AnalyzeAB]

%% R=1, Not Time Sensitive
R=1;
infoState.mean=[mu_vA_P,mu_vB_P];
infoState.stdev=[stdev_vA_P,stdev_vB_P];
infoState.alternatives=[1,2];

%Expectation of selecting immediately A/B
mu_vA_t0=mu_time(mu_vA_P,0);
mu_vB_t0=mu_time(mu_vB_P,0);
stdev_vA_t0=stdev_time(stdev_vA_P,0);
stdev_vB_t0=stdev_time(stdev_vB_P,0);
E_SelectA=utilFunction(mu_vA_t0,stdev_vA_t0,R);
E_SelectB=utilFunction(mu_vB_t0,stdev_vB_t0,R);

%Expectation of testing A
mu_vA_tA1=mu_time(mu_vA_t0,T_A);
```

```
mu_vB_tA=mu_time(mu_vB_t0,T_A);
stdev_vA_tA1=stdev_time(stdev_vA_t0,T_A);
stdev_vB_tA=stdev_time(stdev_vB_t0,T_A);
stdev_epsA=stdev_time(stdev_epsA_P,0);
mu_vA_tA=@(mu_testA)
(mu_vA_tA1*stdev_epsA^2+mu_testA*stdev_vA_tA1^2)/(stdev_epsA^2+stdev_vA
_tA1^2);
stdev_vA_tA=sqrt((stdev_epsA^2*stdev_vA_tA1^2)/(stdev_epsA^2+stdev_vA_t
A1^2));

mu_vA_tAB=@(mu_testA) mu_time((mu_vA_tA(mu_testA)-
c_time(T_A)*V_final)/(1-c_time(T_A)),T_A+T_B);
mu_vB_tAB=mu_time(mu_vB_t0,T_A+T_B);
stdev_vA_tAB=stdev_time(stdev_vA_tA/(1-c_time(T_A)),T_A+T_B);
stdev_vB_tAB=stdev_time(stdev_vB_t0,T_A+T_B);
stdev_epsB=stdev_time(stdev_epsB_P,T_A);
%integral over mu_testA

AA_AB=@(mu_testA)
E_U_Test_X(mu_vA_tAB(mu_testA),mu_vB_tAB,stdev_vA_tAB,stdev_vB_tAB,C_A+
C_B,stdev_epsB,R,2);
AA_SA=@(mu_testA) utilFunction(mu_vA_tA(mu_testA)-C_A,stdev_vA_tA,R);
AA_SB=utilFunction(mu_vB_tA-C_A,stdev_vB_tA,R);
E_AnalyzeA=quadv(@(mu_testA)
max(max(AA_SA(mu_testA),AA_SB),AA_AB(mu_testA)).*normpdf(mu_testA,mu_vA
_tA1,sqrt(stdev_vA_tA1^2+stdev_epsA^2)),mu_vA_tA1-
10*stdev_vA_tA1,mu_vA_tA1+10*stdev_vA_tA1,tol);

%Expectation of testing B
mu_vB_tB1=mu_time(mu_vB_t0,T_B);
mu_vA_tB=mu_time(mu_vA_t0,T_B);
stdev_vB_tB1=stdev_time(stdev_vB_t0,T_B);
stdev_vA_tB=stdev_time(stdev_vA_t0,T_B);
stdev_epsB=stdev_time(stdev_epsB_P,0);
mu_vB_tB=@(mu_testB)
(mu_vB_tB1*stdev_epsB^2+mu_testB*stdev_vB_tB1^2)/(stdev_epsB^2+stdev_vB
_tB1^2);
stdev_vB_tB=sqrt((stdev_epsB^2*stdev_vB_tB1^2)/(stdev_epsB^2+stdev_vB_t
B1^2));

mu_vB_tBA=@(mu_testB) mu_time((mu_vB_tB(mu_testB)-
c_time(T_B)*V_final)/(1-c_time(T_B)),T_A+T_B);
mu_vA_tBA=mu_time(mu_vA_t0,T_A+T_B);
stdev_vB_tBA=stdev_time(stdev_vB_tB/(1-c_time(T_B)),T_A+T_B);
stdev_vA_tBA=stdev_time(stdev_vA_t0,T_A+T_B);
stdev_epsA=stdev_time(stdev_epsA_P,T_B);
%integral over mu_testB

AB_AA=@(mu_testB)
E_U_Test_X(mu_vA_tBA,mu_vB_tBA(mu_testB),stdev_vA_tBA,stdev_vB_tBA,C_A+
C_B,stdev_epsA,R,1);
AB_SB=@(mu_testB) utilFunction(mu_vB_tB(mu_testB)-C_B,stdev_vB_tB,R);
AB_SA=utilFunction(mu_vA_tB-C_B,stdev_vA_tB,R);
E_AnalyzeB=quadv(@(mu_testB)
max(max(AB_SB(mu_testB),AB_SA),AB_AA(mu_testB)).*normpdf(mu_testB,mu_vB
```

```
_tB1,sqrt(stdev_vB_tB1^2+stdev_epsB^2)),mu_vB_tB1-
10*stdev_vB_tB1,mu_vB_tB1+10*stdev_vB_tB1,tol);


%Expectation of testing A and B
if T_A<=T_B
 mu_vA_tA1=mu_time(mu_vA_t0,T_A);
mu_vB_tA=mu_time(mu_vB_t0,T_A);
stdev_vA_tA1=stdev_time(stdev_vA_t0,T_A);
stdev_vB_tA=stdev_time(stdev_vB_t0,T_A);
stdev_epsA=stdev_time(stdev_epsA,0);
stdev_epsB=stdev_time(stdev_epsB,0);
mu_vA_tA=@(mu_testA)
(mu_vA_tA1*stdev_epsA^2+mu_testA*stdev_vA_tA1^2)/(stdev_epsA^2+stdev_vA
_tA1^2);
stdev_vA_tA=sqrt((stdev_epsA^2*stdev_vA_tA1^2)/(stdev_epsA^2+stdev_vA_t
A1^2));

mu_vA_tAB=@(mu_testA) mu_time((mu_vA_tA(mu_testA)-
c_time(T_A)*V_final)/(1-c_time(T_A)),T_B);
mu_vB_tAB=mu_time(mu_vB_t0,T_B);
stdev_vA_tAB=stdev_time(stdev_vA_tA/(1-c_time(T_A)),T_B);
stdev_vB_tAB=stdev_time(stdev_vB_t0,T_B);
%integral over mu_testA

AAB=@(mu_testA)
E_U_Test_X(mu_vA_tAB(mu_testA),mu_vB_tAB,stdev_vA_tAB,stdev_vB_tAB,C_A+
C_B,stdev_epsB,R,2);
E_AnalyzeAB=quadv(@(mu_testA)
AAB(mu_testA).*normpdf(mu_testA,mu_vA_tA1,sqrt(stdev_vA_tA1^2+stdev_eps
A^2)),mu_vA_tA1-10*stdev_vA_tA1,mu_vA_tA1+10*stdev_vA_tA1,tol);

else %T_A>T_B
mu_vB_tB1=mu_time(mu_vB_t0,T_B);
mu_vA_tB=mu_time(mu_vA_t0,T_B);
stdev_vB_tB1=stdev_time(stdev_vB_t0,T_B);
stdev_vA_tB=stdev_time(stdev_vA_t0,T_B);
stdev_epsA=stdev_time(stdev_epsA,0);
stdev_epsB=stdev_time(stdev_epsB,0);
mu_vB_tB=@(mu_testB)
(mu_vB_tB1*stdev_epsB^2+mu_testB*stdev_vB_tB1^2)/(stdev_epsB^2+stdev_vB
_tB1^2);
stdev_vB_tB=sqrt((stdev_epsB^2*stdev_vB_tB1^2)/(stdev_epsB^2+stdev_vB_t
B1^2));

mu_vB_tBA=@(mu_testB) mu_time((mu_vB_tB(mu_testB)-
c_time(T_B)*V_final)/(1-c_time(T_B)),T_A);
mu_vA_tBA=mu_time(mu_vA_t0,T_A);
stdev_vB_tBA=stdev_time(stdev_vB_tB/(1-c_time(T_B)),T_A);
stdev_vA_tBA=stdev_time(stdev_vA_t0,T_A);
%integral over mu_testB

ABA=@(mu_testB)
E_U_Test_X(mu_vA_tBA,mu_vB_tBA(mu_testB),stdev_vA_tBA,stdev_vB_tBA,C_A+
C_B,stdev_epsA,R,1);
```

```
E_AnalyzeAB=quadv(@(mu_testB)
ABA(mu_testB).*normpdf(mu_testB,mu_vB_tB1,sqrt(stdev_vB_tB1^2+stdev_eps
B^2)),mu_vB_tB1-10*stdev_vB_tB1,mu_vB_tB1+10*stdev_vB_tB1,tol);
end

dec_R1_c0=[E_SelectA;E_SelectB;E_AnalyzeA;E_AnalyzeB;E_AnalyzeAB]
```

E_U_test_X.m

```
function =E_U_Test_X(mu_uA,mu_uB,stdev_uA,stdev_uB,cost,
stdev_eps,R,TestX)

%% E_test=E_U_Test(mu_uA,mu_uB,stdev_uA,stdev_uB,cost,stdev_eps,R)
%  E_U_Test_X() is a function that algebraically calculates the
expected
%  value of performing a test on design alternative X defined by
'TestX'
%  If the test were
%  performed, a test value V_TX would be returned, and the prior
beliefs
%  about the distribution of the utility of the alternative is updated
%  using Bayesian statistics.  An integral over all possible test
values
%  was calculated to generate the expected utility of performing the
test.
%
%
%  **Parameters**
%   Priors (mu_uA,mu_uB,stdev_uA,stdev_uB) of the distribution of the
%      possible values for two design alternatives 'A' and 'B'
%   Information about the cost 'cost' of testing the alternative and
%      information about the quality of the test characterized by
%      its standard deviation 'stdev_eps.'
%   The constant risk aversion 'R' of the decision maker.
%   A definition variable 'TestX' which is 1 if X='A' is the
alternative being
%   tested, and 2 if X='B' is the alternative being tested.
%%  Variable Definition
if TestX==1  %A is being tested
    mu_uX=mu_uA;
    mu_uY=mu_uB;
    stdev_uX=stdev_uA;
    stdev_uY=stdev_uB;
elseif TestX==2  %B is being tested
    mu_uX=mu_uB;
    mu_uY=mu_uA;
    stdev_uX=stdev_uB;
    stdev_uY=stdev_uA;
else
    error('TestX must be either 1 for A or 2 for B.')
end
```

```
%% Intermediate calculations

stdev_Vtx=sqrt(stdev_uX^2+stdev_eps^2);
    %Standard deviation for distribution of uX given the value of the
test
stdev_XgivenLQ=sqrt(stdev_uX^2*stdev_eps^2/stdev_Vtx^2);
    %Critical test value at which the E(uX)=E(uY) after testing X
V_TXStar=(1/stdev_uX^2)*(stdev_Vtx^2*mu_uY-
stdev_eps^2*mu_uX+R/2*(stdev_uX^2*stdev_eps^2-stdev_Vtx^2*stdev_uY^2));

%% Define Expectations for all risk considerations
if R==0
    E_test=(mu_uY-cost).*normcdf(V_TXStar,mu_uX,stdev_Vtx)+...
        (stdev_eps^2*mu_uX/stdev_Vtx^2-cost).*(1-
normcdf(V_TXStar,mu_uX,stdev_Vtx))+...
        stdev_uX^2/(sqrt(2*pi)*stdev_Vtx^2)*(stdev_Vtx*exp(-(V_TXStar-
mu_uX).^2/(2*stdev_Vtx^2))+mu_uX*sqrt(pi/2).*(2-2*normcdf((V_TXStar-
mu_uX)/(stdev_Vtx))));
else
    E_test=1/R*(1-...
        (exp(-R*(mu_uY-cost)+1/2*stdev_uY^2*R^2))...
        .*normcdf((V_TXStar-mu_uX)/stdev_Vtx)...
        -exp(1/2*stdev_XgivenLQ^2*R^2+R*(R*stdev_uX^4/(2*stdev_Vtx^2)-
mu_uX+cost))...
        .*(1-normcdf((V_TXStar+R*stdev_uX^2-mu_uX)/stdev_Vtx)));
end

if isinf(mu_uY)
    if R==0
        E_test=(stdev_eps^2*mu_uX/stdev_Vtx^2-cost).*(1-
normcdf(V_TXStar,mu_uX,stdev_Vtx))+...
            stdev_uX^2/(sqrt(2*pi)*stdev_Vtx^2)*(stdev_Vtx*exp(-
(V_TXStar-mu_uX).^2/(2*stdev_Vtx^2))+mu_uX*sqrt(pi/2).*(2-
2*normcdf((V_TXStar-mu_uX)/(stdev_Vtx))));
    else
        E_test=1/R*(1-
exp(1/2*stdev_XgivenLQ^2*R^2+R*(R*stdev_uX^4/(2*stdev_Vtx^2)-
mu_uX+cost)));
    end
end
```

Boundary_Plot_Time

```
% Boundary_Plot_Time is a script that helps to visualize the regions
where
% a single decision alternative provides the maximum utility.  The
decision
% alternatives available are:
% Select A
% Analyze A
% Analyze A&B at the same time
```

```
% Analyze B
% Select B
% The analysis performed is similar to that of SAE_Single_scenario1.m,
but
% iterated for multiple points and then common areas are plotted into
% boundary plots.

% mu_vA_P - Prior Belief about the mean of the value of A
% mu_vB_P - Prior Belief about the mean of the value of B
% stdev_vA_P - Prior Belief about the standard deviation of the value
of A
% stdev_vB_P - Prior Belief about the standard deviation of the value
of B
% R - Risk coefficient
% C_A - The cost of Analyzing A
% C_B - The cost of Analyzing B
% stdev_epsA - A term describing the accuracy of the Analysis performed
on A
% stdev_epsB - A term describing the accuracy of the Analysis performed
on B
% T_A - The time required to perform the Analysis of A (0 = no time)
% T_B - The time required to perform the Analysis of B (1 = a long
time)

clear
clc
close all

resolution=30;

addpath C:\Users\Ben\Documents\RiskTimeAnalysis\Functions
load c_time_vec.mat
c_time=@(t) c_time(t,c_t);
%% Define Parameters
R=0;
C_A=.016;
C_B=.016;
stdev_epsA_P=.07;
stdev_epsB_P=.07;
T_A=2/3;     %in current format T_A<1,T_B<1,
T_B=2/3;
mu_vA_Pvec=linspace(-5,5,resolution);
mu_vB_P=0;
stdev_vA_Pvec=linspace(0,2,resolution);
stdev_vB_P=1;
tol=1e-6;
%preallocate for time concerns
E_SelectA=zeros(length(mu_vA_Pvec),length(stdev_vA_Pvec));
E_SelectB=zeros(length(mu_vA_Pvec),length(stdev_vA_Pvec));
E_AnalyzeA=zeros(length(mu_vA_Pvec),length(stdev_vA_Pvec));
E_AnalyzeB=zeros(length(mu_vA_Pvec),length(stdev_vA_Pvec));
E_AnalyzeAB=zeros(length(mu_vA_Pvec),length(stdev_vA_Pvec));

%Define value as a function of time
% val_time=V_init-(V_init-V_final)*c(t)
```

```
%   Because val_init is a normal distribution, the mean must be altered
by the coefficient c_time.
V_final=-5;
mu_time=@(v_init,time) v_init*(1-c_time(time))+V_final*c_time(time);
stdev_time=@(stdev_init,time) (1-c_time(time))*stdev_init;



for a=1:length(mu_vA_Pvec);
    for b=1:length(stdev_vA_Pvec);
        %Define initial beliefs
        mu_vA_P=mu_vA_Pvec(a);
        stdev_vA_P=stdev_vA_Pvec(b);

        infoState.mean=[mu_vA_P,mu_vB_P];
        infoState.stdev=[stdev_vA_P,stdev_vB_P];
        infoState.alternatives=[1,2];

        %Expectation of selecting immediately A/B
        mu_vA_t0=mu_time(mu_vA_P,0);
        mu_vB_t0=mu_time(mu_vB_P,0);
        stdev_vA_t0=stdev_time(stdev_vA_P,0);
        stdev_vB_t0=stdev_time(stdev_vB_P,0);
        E_SelectA(a,b)=utilFunction(mu_vA_t0,stdev_vA_t0,R);
        E_SelectB(a,b)=utilFunction(mu_vB_t0,stdev_vB_t0,R);

        %Expectation of testing A
        mu_vA_tA1=mu_time(mu_vA_t0,T_A);
        mu_vB_tA=mu_time(mu_vB_t0,T_A);
        stdev_vA_tA1=stdev_time(stdev_vA_t0,T_A);
        stdev_vB_tA=stdev_time(stdev_vB_t0,T_A);
        stdev_epsA=stdev_time(stdev_epsA_P,0);
        mu_vA_tA=@(mu_testA)
(mu_vA_tA1*stdev_epsA^2+mu_testA*stdev_vA_tA1^2)/(stdev_epsA^2+stdev_vA
_tA1^2);

stdev_vA_tA=sqrt((stdev_epsA^2*stdev_vA_tA1^2)/(stdev_epsA^2+stdev_vA_t
A1^2));

        mu_vA_tAB=@(mu_testA) mu_time((mu_vA_tA(mu_testA)-
c_time(T_A)*V_final)/(1-c_time(T_A)),T_A+T_B);
        mu_vB_tAB=mu_time(mu_vB_t0,T_A+T_B);
        stdev_vA_tAB=stdev_time(stdev_vA_tA/(1-c_time(T_A)),T_A+T_B);
        stdev_vB_tAB=stdev_time(stdev_vB_t0,T_A+T_B);
        stdev_epsB=stdev_time(stdev_epsB_P,T_A);
        %integral over mu_testA

        AA_AB=@(mu_testA)
E_U_Test_X(mu_vA_tAB(mu_testA),mu_vB_tAB,stdev_vA_tAB,stdev_vB_tAB,C_A+
C_B,stdev_epsB,R,2);
        AA_SA=@(mu_testA) utilFunction(mu_vA_tA(mu_testA)-
C_A,stdev_vA_tA,R);
        AA_SB=utilFunction(mu_vB_tA-C_A,stdev_vB_tA,R);
        E_AnalyzeA(a,b)=quadv(@(mu_testA)
max(max(AA_SA(mu_testA),AA_SB),AA_AB(mu_testA)).*normpdf(mu_testA,mu_vA
```

```
_tA1,sqrt(stdev_vA_tA1^2+stdev_epsA^2)),mu_vA_tA1-
10*stdev_vA_tA1,mu_vA_tA1+10*stdev_vA_tA1,tol);

        %Expectation of testing B
        mu_vB_tB1=mu_time(mu_vB_t0,T_B);
        mu_vA_tB=mu_time(mu_vA_t0,T_B);
        stdev_vB_tB1=stdev_time(stdev_vB_t0,T_B);
        stdev_vA_tB=stdev_time(stdev_vA_t0,T_B);
        stdev_epsB=stdev_time(stdev_epsB_P,0);
        mu_vB_tB=@(mu_testB)
(mu_vB_tB1*stdev_epsB^2+mu_testB*stdev_vB_tB1^2)/(stdev_epsB^2+stdev_vB
_tB1^2);

stdev_vB_tB=sqrt((stdev_epsB^2*stdev_vB_tB1^2)/(stdev_epsB^2+stdev_vB_t
B1^2));

        mu_vB_tBA=@(mu_testB) mu_time((mu_vB_tB(mu_testB)-
c_time(T_B)*V_final)/(1-c_time(T_B)),T_A+T_B);
        mu_vA_tBA=mu_time(mu_vA_t0,T_A+T_B);
        stdev_vB_tBA=stdev_time(stdev_vB_tB/(1-c_time(T_B)),T_A+T_B);
        stdev_vA_tBA=stdev_time(stdev_vA_t0,T_A+T_B);
        stdev_epsA=stdev_time(stdev_epsA_P,T_B);
        %integral over mu_testB

        AB_AA=@(mu_testB)
E_U_Test_X(mu_vA_tBA,mu_vB_tBA(mu_testB),stdev_vA_tBA,stdev_vB_tBA,C_A+
C_B,stdev_epsA,R,1);
        AB_SB=@(mu_testB) utilFunction(mu_vB_tB(mu_testB)-
C_B,stdev_vB_tB,R);
        AB_SA=utilFunction(mu_vA_tB-C_B,stdev_vA_tB,R);
        E_AnalyzeB(a,b)=quadv(@(mu_testB)
max(max(AB_SB(mu_testB),AB_SA),AB_AA(mu_testB)).*normpdf(mu_testB,mu_vB
_tB1,sqrt(stdev_vB_tB1^2+stdev_epsB^2)),mu_vB_tB1-
10*stdev_vB_tB1,mu_vB_tB1+10*stdev_vB_tB1,tol);



        %Expectation of testing A and B
        if T_A<=T_B
         mu_vA_tA1=mu_time(mu_vA_t0,T_A);
        mu_vB_tA=mu_time(mu_vB_t0,T_A);
        stdev_vA_tA1=stdev_time(stdev_vA_t0,T_A);
        stdev_vB_tA=stdev_time(stdev_vB_t0,T_A);
        stdev_epsA=stdev_time(stdev_epsA,0);
        stdev_epsB=stdev_time(stdev_epsB,0);
        mu_vA_tA=@(mu_testA)
(mu_vA_tA1*stdev_epsA^2+mu_testA*stdev_vA_tA1^2)/(stdev_epsA^2+stdev_vA
_tA1^2);

stdev_vA_tA=sqrt((stdev_epsA^2*stdev_vA_tA1^2)/(stdev_epsA^2+stdev_vA_t
A1^2));

        mu_vA_tAB=@(mu_testA) mu_time((mu_vA_tA(mu_testA)-
c_time(T_A)*V_final)/(1-c_time(T_A)),T_B);
        mu_vB_tAB=mu_time(mu_vB_t0,T_B);
        stdev_vA_tAB=stdev_time(stdev_vA_tA/(1-c_time(T_A)),T_B);
        stdev_vB_tAB=stdev_time(stdev_vB_t0,T_B);
```

```
        %integral over mu_testA

        AAB=@(mu_testA)
E_U_Test_X(mu_vA_tAB(mu_testA),mu_vB_tAB,stdev_vA_tAB,stdev_vB_tAB,C_A+
C_B,stdev_epsB,R,2);
        E_AnalyzeAB(a,b)=quadv(@(mu_testA)
AAB(mu_testA).*normpdf(mu_testA,mu_vA_tA1,sqrt(stdev_vA_tA1^2+stdev_eps
A^2)),mu_vA_tA1-10*stdev_vA_tA1,mu_vA_tA1+10*stdev_vA_tA1,tol);

        else %T_A>T_B
        mu_vB_tB1=mu_time(mu_vB_t0,T_B);
        mu_vA_tB=mu_time(mu_vA_t0,T_B);
        stdev_vB_tB1=stdev_time(stdev_vB_t0,T_B);
        stdev_vA_tB=stdev_time(stdev_vA_t0,T_B);
        stdev_epsA=stdev_time(stdev_epsA,0);
        stdev_epsB=stdev_time(stdev_epsB,0);
        mu_vB_tB=@(mu_testB)
(mu_vB_tB1*stdev_epsB^2+mu_testB*stdev_vB_tB1^2)/(stdev_epsB^2+stdev_vB
_tB1^2);

stdev_vB_tB=sqrt((stdev_epsB^2*stdev_vB_tB1^2)/(stdev_epsB^2+stdev_vB_t
B1^2));

        mu_vB_tBA=@(mu_testB) mu_time((mu_vB_tB(mu_testB)-
c_time(T_B)*V_final)/(1-c_time(T_B)),T_A);
        mu_vA_tBA=mu_time(mu_vA_t0,T_A);
        stdev_vB_tBA=stdev_time(stdev_vB_tB/(1-c_time(T_B)),T_A);
        stdev_vA_tBA=stdev_time(stdev_vA_t0,T_A);
        %integral over mu_testB

        ABA=@(mu_testB)
E_U_Test_X(mu_vA_tBA,mu_vB_tBA(mu_testB),stdev_vA_tBA,stdev_vB_tBA,C_A+
C_B,stdev_epsA,R,1);
        E_AnalyzeAB(a,b)=quadv(@(mu_testB)
ABA(mu_testB).*normpdf(mu_testB,mu_vB_tB1,sqrt(stdev_vB_tB1^2+stdev_eps
B^2)),mu_vB_tB1-10*stdev_vB_tB1,mu_vB_tB1+10*stdev_vB_tB1,tol);
        end
    end
end
E_AB=E_AnalyzeAB;
E_AB(:,1)=E_AB(:,2);
%% Plot results

[STD_STD,MU_MU]=meshgrid(stdev_vA_Pvec,mu_vA_Pvec);
color=6*ones(size(MU_MU));
surf(MU_MU,STD_STD,E_SelectA,color)
hold on
color(:,:)=6;
surf(MU_MU,STD_STD,E_SelectB,color)
color(:,:)=3;
surf(MU_MU,STD_STD,E_AnalyzeA,color)
color(:,:)=4;
surf(MU_MU,STD_STD,E_AnalyzeB,color)
color(:,:)=5;
surf(MU_MU,STD_STD,E_AB,color)
colormap(gray(10))
```

```
axis([min(mu_vA_Pvec) max(mu_vA_Pvec) min(stdev_vA_Pvec)
max(stdev_vA_Pvec) -10 10])
shading flat
% xlabel('\mu_A')
% ylabel('\sigma_A')
% zlabel('E(decision)')
% legend('SelectA','SelectB','AnalyzeA','AnalyzeB','AnalyzeAB')

% title(sprintf('Decision Analysis for R=%d, C_A=%d, C_B=%d,
Stdev_e_p_s_A=%d,Stdev_e_p_s_B=%d',R,C_A,C_B,stdev_epsA,stdev_epsB))
view(2)
box on

% %% refine plot results
% MATRIX(:,:,1)=E_SelectA;
% MATRIX(:,:,2)=E_SelectB;
% MATRIX(:,:,3)=E_AnalyzeA;
% MATRIX(:,:,4)=E_AnalyzeB;
% MATRIX(:,:,5)=E_AnalyzeAB;
%
% for ii=1:resolution
%     for jj=1:resolution
%         best=max(MATRIX(ii,jj,:));
%
%             MATRIX(ii,jj,find(MATRIX(ii,jj,:)<best))=NaN;
%     end
% end
%
%
% [STD_STD,MU_MU]=meshgrid(stdev_vA_Pvec,mu_vA_Pvec);
% color=5*ones(size(MU_MU));
% plot3(MU_MU,STD_STD,MATRIX(:,:,1),'k.')
% hold on
% color(:,:)=5;
% plot3(MU_MU,STD_STD,MATRIX(:,:,2),'ko')
% color(:,:)=5;
% % plot3(MU_MU,STD_STD,MATRIX(:,:,3),'k+')
% color(:,:)=5;
% % plot3(MU_MU,STD_STD,MATRIX(:,:,4),'k.')
% color(:,:)=5;
% % plot3(MU_MU,STD_STD,MATRIX(:,:,5),'k.')
% colormap(gray(6))
% axis([min(mu_vA_Pvec) max(mu_vA_Pvec) min(stdev_vA_Pvec)
max(stdev_vA_Pvec) -10 10])
% shading flat
% xlabel('mu')
% ylabel('stdev')
% zlabel('E(decision)')
% legend('SelectA','SelectB','AnalyzeA','AnalyzeB','AnalyzeAB')
%
% title(sprintf('Decision Analysis for R=%d, C_A=%d, C_B=%d,
Stdev_e_p_s_A=%d,Stdev_e_p_s_B=%d',R,C_A,C_B,stdev_epsA,stdev_epsB))
% view(2)
%
% plot3(-1,.47,5,'k*')
```

REFERENCES

[1] Thompson, S.C. and Paredis, C.J.J. *A Process-Centric Problem Formulation for Decision-Based Design*. in *ASME IDETC/CIE 2009*. 2009. San Diego, California, USA.

[2] Hazelrigg, G.A., *A Framework for Decision-Based Engineering Design.* ASME Journal of Mechanical Design, 1998. **120**: p. 653-658.

[3] Gero, J., *Design Prototypes: A Knowledge Representation Schema for Design*, in *AI Magazine*. 1990. p. 26-36.

[4] Tomiyama, T. and Yoshikawa, H., *Extended general design theory.* 1986.

[5] Braha, D. and Reich, Y., *Topological structures for modeling engineering design processes.* Research in Engineering Design, 2003. **14**(4): p. 185-199.

[6] Umeda, Y., et al., *Function, behaviour, and structure.* Applications of Artificial Intelligence in Engineering V, 1990. **1**: p. 177-193.

[7] Suh, N.P., *The principles of design*. 1990: Oxford University Press, USA.

[8] Lewis, K., Chen, W., and Schmidt, L.C., eds. *Decision Making in Engineering Design*. 2006, American Society of Mechanical Engineers: New York. 344.

[9] von Neumann, J. and Morgenstern, O., *Theory of Games and Economic Behavior*. 1944, Princeton, NJ: Princeton University Press.

[10] Mehr, A.F. and Tumer, I.Y., *Risk-Based Decision-Making for Managing Resources During the Design of Complex Space Exploration Systems.* Journal of Mechanical Design, 2006. **128**(4): p. 1014-1022.

[11] Berger, J.O., *Statistical Decision Theory and Bayesian Analysis*. 2nd ed. 1985, New York: Springer.

[12] Luce, R.D. and Raiffa, H., *Games and Decisions*. 1957, New York: Wiley.

[13] Herstein, I. and Milnor, J., *An axiomatic approach to measurable utility.* Econometrica, 1953. **21**(2): p. 291-297.

[14] Marschak, J., *Rational Behavior, Uncertain Prospects, and Measurable Utility.* Econometrica, 1950. **18**(2): p. 111-141.

[15] Keeney, R.L. and Raiffa, H., *Decisions with Multiple Objectives*. 2 ed. 1993, Cambridge, UK: Cambridge University Press. 569.

[16] Keeney, R.L., *Decision Analysis: An Overview*. Operations Research, 1982. **30**(5): p. 803-838.

[17] Thurston, D.L., *Real and Misconcieved Limitations to Decision Based Design with Utility Analysis.* Journal of Mechanical Design, 2001. **123**: p. 176-182.

[18] Taguchi, G., *Introduction to Quality Engineering*. 1986, Tokyo: Asian Productivity Organization.

[19] Taguchi, G. and Clausing, D., *Robust quality.* Harvard Business Review, 1990. **68**(1): p. 65-75.

[20] Nair, V.N., *Taguchi's Parameter Design: A Panel Discussion.* Technometrics, 1992. **34**: p. 127-161.

[21] Chen, W., et al., *A Procedure for Robust Design: Minimizing Variations Caused by Noise Factors and Control Factors.* ASME Journal of Mechanical Design, 1996. **118**(4): p. 478-485.

[22] Bras, B. and Mistree, F., *Robust Design using Compromise Decision Support Problems.* Engineering Optimization, 1993. **21**: p. 213-239.
[23] Chen, W., Wiecek, M.M., and Zhang, J., *Quality Utility - A Compromise Programming Approach to Robust Design.* Journal of Mechanical Design, 1999. **121**(2): p. 179-187.
[24] Chen, W., et al., *Exploration of the Effectiveness of Physical Programming in Robust Design.* ASME Journal of Mechanical Design, 2000. **122**(2): p. 155-163.
[25] Rao, S.S., *Reliability-based design*. 1992: McGraw-Hill, New York.
[26] Chandu, S.V.L. and Grandhi, R.V., *General purpose procedure for reliability based structural optimization under parametric uncertainties.* Advances in Engineering Software, 1995. **23**(1): p. 7-14.
[27] Enevoldsen, I., *Reliability-based optimization as an information tool.* Mechanics Based Design of Structures and Machines, 1994. **22**(1): p. 117-135.
[28] Wu, Y.T., *Computational Methods for Efficient Structural Reliability and Reliability Sensitivity Analysis.* AIAA Journal, 1994. **32**(8): p. 1717-1732.
[29] Rackwitz, R., *Reliability analysis--a review and some perspectives.* Structural Safety, 2001. **23**(4): p. 365-395.
[30] Qiu, Y., Ge, P., and Yim, S.C., *Risk-based Resource Allocation for Collaborative System Design in a Distributed Environment.* Journal of Mechanical Design, 2008. **130**(6).
[31] Feather, M. and Cornford, S., *Quantitative Risk-Based Requirements Reasoning.* Requirements Engineering, 2003. **8**(4): p. 248-265.
[32] Tumer, I.Y., Barrientos, F.A., and Mehr, A.F. *Towards risk based design (RBD) of space exploration missions: A review of RBD practice and research trends at NASA*. 2005. New York, NY 10016-5990, United States: American Society of Mechanical Engineers.
[33] Motte, D. *Study of Alternative Strategies to the Task Clarification Activity of the Market-Pull Product Development Process Model*. in *ASME IDETC/CIE 2009*. 2009. San Diego, California, USA.
[34] Pawar, K.S., Menon, U., and Riedel, J., *Time to market.* Integrated manufacturing systems, 1994. **5**(1): p. 14-22.
[35] Urban, G.L., et al., *Market share rewards to pioneering brands: An empirical analysis and strategic implications.* Management Science, 1986: p. 645-659.
[36] Reinertsen, D.G. *The Economics of Gaining Competitive Advantage through Development Speed*. in *IEEE International Engineering Management Conference*. 1990.
[37] Reinertsen, D.G., *Whodunit? The Search for the New-Product Killers*, in *Electronic Business*. 1983.
[38] Charney, C., *Time to market: reducing product lead time*. 1991: Society of Manufacturing.
[39] Raiffa, H., *Decision analysis*. 1968: Addison-Wesley Reading, MA.