Risk-Based Decision-Making for Managing Resources during the Design of Complex Aerospace Systems

Ali Farhang Mehr, Ph.D.
amehr@email.arc.nasa.gov

Irem Y. Tumer, Ph.D.
itumer@mail.arc.nasa.gov

Complex Systems Design Group
Intelligent Systems Division
NASA Ames Research Center
Moffett Field, CA 94035 USA

ABSTRACT

Complex aerospace systems are often designed in collaborative engineering environments where requirements and design decisions are rapidly made and withdrawn by various subsystems as the process evolves. As a result, the system-level management should continuously re-allocate resources (e.g. capital and labor) among various subsystems such that the main objectives of the system are achieved as closely as possible. *Minimizing risk* has been long accepted as one of the major criterion for system-level decisions and particularly resource management. In this context, *Risk-based Decision Making* refers to a process that allocates resources in such a way that the expected risk of the overall system is minimized. While a variety of quantitative and qualitative techniques to mitigate risk and uncertainty have been developed over the years, they mostly aim at the final stages of the design process only, and therefore are not applicable to the early stages of design. In particular, since the early conceptual design is often conducted by concurrent (and sometimes distributed) engineering teams, most risk management methods cannot effectively and dynamically capture the risk contribution of various design decisions in order to make risk-optimal resource allocation decisions accordingly. As such, this paper presents a new risk-based design decision-making method, referred to as *Risk and Uncertainty Based Concurrent Integrated Design Methodology* or *RUBIC Design Methodology* for short. The new approach is based on concepts from portfolio optimization theory and continuous resource management, extended to provide a mathematical rigor for risk-based decision-making in concurrent engineering environments, where various subsystem experts, computer simulations, theoretical models, and historical design failure databases are involved. The RUBIC design method is based on the idea that a unit of resources that is allocated to reduce the level of risk at a certain component contributes to the overall system risk reduction in the following two ways: 1) by reducing the risk premium of the component itself (the exact amount of reduction in the risk premium is not fully known a priori, but can be estimated as a probability distribution based on historical or theoretical models); and 2) by impacting the risk premiums of other components (i.e. the correlation among components). RUBIC then provides a probabilistic framework for reducing the expected risk of the final engineering system via optimal allocation of available risk-mitigation resources. The application of the proposed approach to both single-subsystem and multi-subsystem design processes is demonstrated using a satellite reaction wheel example.

1. PROLOGUE

Because of the complex and multi-subsystem nature of aerospace systems, they are always designed in a multi-subsystem hierarchical structure with several domain experts and system-level managers. Each subsystem has a subsystem chair that assumes the responsibility for that subsystem, while most high-level goals and constraints are determined at the system-level based on the overall requirements and constraints of the project. Often, the variables, constraints and goals are only loosely defined in both system and subsystem levels. In such environments, design decisions are often made and reversed in both system and subsystem levels as the design process converges. Therefore, the main job of the system-level managers is to re-allocate available resources to guide the project in a desirable direction. *Risk*, as one of the major drivers for system-level decisions, plays an important role in this decision-making process. Researchers, therefore, have developed a wide variety of risk and failure identification methods over the past few decades to enable a more objective *risk-based design decision-making* process (For review of such methods, see for instance, Schrader et al. 1993, Zang et al. 2000, Backman 2000, Choi 2001, Du and Chen 2002, Smith and Mahadevan 2003). In particular, failure analysis tools have been widely used to evaluate the safety of aerospace systems by identifying ways to reduce the likelihood of failure through design changes. Examples of the most commonly used methods are: Failure Modes Effects Analysis (FMEA), Fault Tree Analysis (FTA) and Event Tree Analysis (ETA). These basic techniques continue to evolve and have spawned other techniques such as the Failure Modes and Critical Effects Analysis (FMECA), Event Sequence Diagrams (ESD), Reliability Block Diagrams (RBD) and the Master Logic Diagram (MLD) (See Greenfield 2000 for a review of common risk analysis techniques in the aerospace industry).

While failure analysis tools provide ways to identify potential risks, their connection to system-level decisions (specifically, resource allocation) is mostly ad-hoc. In addition, most of these techniques target a certain stage of the design process and cannot be used for *continuous resource management* during the entire span of conceptual design to final design. In particular, the bulk of these techniques fall short in the earlier stages of designing complex aerospace systems where concurrent engineering teams are involved in making rapid design decisions in a hierarchical multi-subsystem design architecture. This is despite the fact that our work, as well as many others studies, have pointed to the early design stages as one of the best opportunities to

catch potential failures and anomalies (e.g. Smith and Mahadevan 2003, Tumer and Stone 2003). Therefore, this paper describes a new approach to risk-based decision-making in the concurrent design of aerospace system. The proposed methodology, hereafter referred to as *Risk and Uncertainty Based Integrated Concurrent* Design Methodology *or RUBIC* for short, has several characteristics that make it a desirable risk-based design decision making tool for complex aerospace systems (and alike, e.g. large-scale information systems).

- RUBIC accounts for both individual *risk premiums* of a component or functional element as well as the impact of reducing the potential risk of one component on its neighbouring elements (i.e. *risk correlation* among elements)

- RUBIC calculates the optimal allocation of resources among subsystems and components such that the expected total risk of the final system is minimized (high-level objective).

- RUBIC enables making decisions in a *real-time and dynamic* fashion such that as the design process evolves, the management can quickly re-calculate its position and make near-optimal decisions. That is, given enough computational power, the RUBIC design methodology provides a real-time and evolving *resource allocation vector* (described later in the paper) that can be used to mitigate risks throughout the design process and in both system and subsystem levels. In this context, RUBIC can be considered a risk-based *continuous resource management* tool that spans the entire design cycle.

The organization of the rest of this paper is as follows: Section 2 provides a brief overview of functional models and how they can be used to represent the overall as the design process evolves. The RUBIC design method employs functional models to identify risks (at the functional level) and allocate mitigation resources accordingly[1]. The new approach of this paper, i.e. RUBIC design methodology, is described in Section 3 for both single-subsystem and multi-subsystem cases. The design example of Figure 1 is used to demonstrate the application of the proposed approach. In Section 4, we discuss ways that functional failure rates can be estimated. We will particularly focus on method to derive numerical estimations from a historical failure database. Section 5 is devoted to the application of RUBIC to the specific problem of designing space exploration systems at various NASA centers. Finally, Section 6 provides the concluding remarks of the paper.

2. FUNCTIONAL MODELING IN MULTI-LEVEL AEROSPACE SYSTEMS

The RUBIC design method employs functional models to represent the state of the design process. This is particularly helpful during the conceptual design phase, where the design has not yet converged to its final form and the component information is often either vague or not available (Stone and Wood 2000, Hirtz 2001). Functional information, in contrast, can be easily derived from the project requirements and decomposed and distributed top-down to the subsystems. In fact, the early stages of the design process can be best described using functional modeling methods that provide a description of the final product as a system of elementary functions that will collectively achieve the overall system-level goals (See, for instance, Hunt et al. 1995, Stone et al. 2000, Stone and Wood 2000, Hirtz et al. 2001). A functional model is basically a flow diagram that shows different functional elements as well as the flow of energy, material, and information through these elements. The RUBIC design methodology reallocates resources as the functional model evolves to satisfy all design requirements and constraints.

Figure 1 shows the functional model of a design example that will be used later in this paper to demonstrate the application of the proposed approach. This design problem involves a satellite reaction wheel, which is a motorized flywheel that can adjust its spin rate to control the positioning of a satellite. As the motor speeds up or slows down, it generates a reacting torque on the body of the satellite that can be used to position the spacecraft. Figure 1 depicts a high-level functional model of a reaction wheel, at a certain point in its design cycle. This design consists of 4 main sub-systems (shaded differently in Figure 1): 1-Motor Controller Subsystem; 2-Motor Subsystem; 3-Flywheel Subsystem; and 4-Structure Subsystem. Note that the reaction wheel as a whole can be considered a subsystem for the overall system, i.e., satellite, which indicates the multi-level nature of designing complex engineering systems.

[1] We will also state that the RUBIC design methodology is more general in principle and can be readily extended and generalized to other forms of modeling the design process (other than functional modeling).

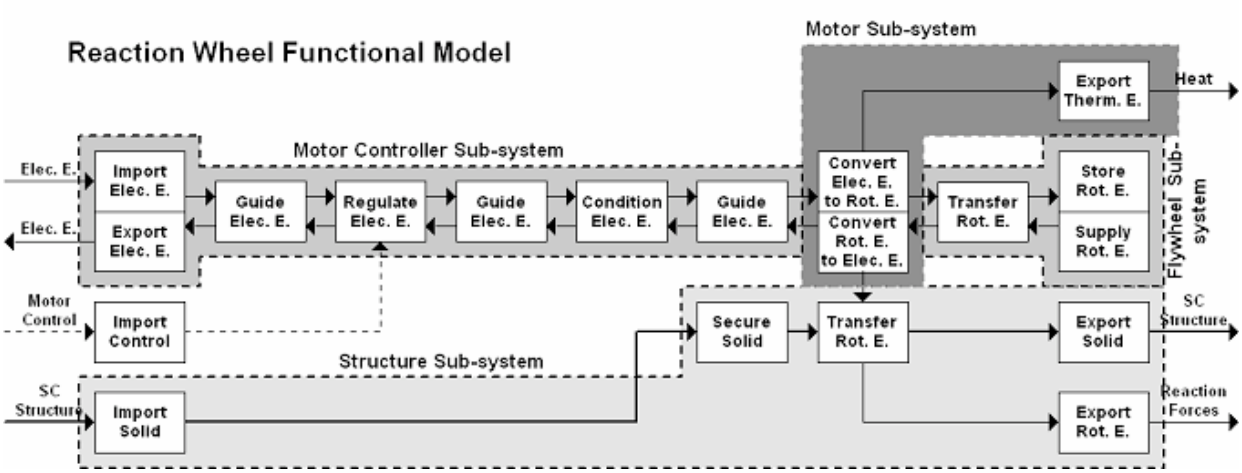

Figure 1 – A high-level functional model of a satellite reaction wheel at some point in its conceptual design phase. A satellite reaction wheel is used to position spacecrafts in the desired direction. Four major subsystems can be identified in this design (distinguished using different shades in the figure).

We use functional models in RUBIC mainly because they can be used throughout the design cycle (from the initial conceptual design to the final design). At each point, the current state of the design is represented by its associated functional model (which evolves and converges to more detailed lower-level models as time progresses). In the next section, we will link the risk and uncertainty of a given design to the estimated functional failure rate at each functional element. Finally, it should be noted that although we chose to use functional models to increase the applicability and lifecycle of the proposed RUBIC design approach, the methodology itself is more general in principle and can be used in a very similar fashion based on the physical components of the system (rather than their functions).

3. RUBIC DESIGN METHODOLGOY

The proposed methodology of this paper is described in this section for both single-subsystem and multi-subsystem cases.

3.1 Assumptions

A functional failure is defined as an undesirable process which results in one (or more) functional elements performing a function other than what it was originally designed for. The probability of a functional failure can be obtained from either: 1- historical data; 2- engineers' intuition; or 3- reasonable estimations and bounds obtained from fundamental principles or

computer simulations (more on computing these rates in Section 4). In a given design, each functional element is designed to perform a task that contributes to the overall success of the system and has an associated *risk premium* (non-negative probability of failure), i.e.

Postulate: Each functional element in a complex system creates a *risk premium*. Two types of risk premiums can then be identified:

1-Insured Risk: A risky element whose risk premium is balanced off with another element (e.g., a sensor to detect failure or a redundant part). In other words, the risk of the original element is balanced off by another element (such as a sensor or a redundant part).

2- Uninsured Risk: A risky element with an unbalanced risk premium (i.e., an element with a known probability of failure but without a risk-balancing element to offset the risk).
In this context, the main goal of the RUBIC design method is to address the following problem:

RUBIC Design Methodology: The RUBIC design methodology is a *continuous risk management* technique in that it identifies the risk elements during the conceptual design phase and continuously optimizes resources (e.g. capital, labor) to mitigate those risks. In other words, the RUBIC design methodology allocates resources to either reduce the risk premium of individual elements (by redesigning the physical components), or balance those risks against other elements (for example by adding redundancies).

We also assume that the only thing that matters to the managers is the total amount of risk in the system. In other words, we assume that risk can be traded homogenously between elements and subsystems. This is a rather restraining but common assumption (e.g., Greenfield 2000). This assumption, however, can be somewhat relaxed by weighting the criticality of different elements and subsystems (which is not the focus of this research).

Postulate: Risk can be traded homogeneously between subsystems and elements.

We also assume that:

Postulate: Risk of an element is not independent of other elements in its subsystem. (This is handled in RUBIC via a covariance matrix). However, the risk of elements in one subsystem is independent of those in another subsystem. Example: The risk of a tank regulating valve in the propulsion subsystem is not independent of the risk of the pressurized Helium tank it is attached to. However, the risk of that same valve is independent of the risk of a seal in another subsystem.

Finally, It is implied in the RUBIC design methodology that a risk can be actually reduced by allocating resources such as time, money, or computational resources (although the amount of risk reduction is not known a priori and will be modeled using a stochastic process in the next section):

Postulate: *Risk* can be traded for "*Risk-Mitigation Resource*". For instance, in the early stages of design, one can conduct a risk reduction study of a certain functional (or physical) element and design safeguards or make design modifications to reduce risk.

In other words, by consuming such resources (e.g., time, cost, computational resource etc.), one can find ways to mitigate the risk of a certain functional (or physical) element, although the actual amount of risk reduction is not known beforehand and should be modeled using a random distribution. This is discussed in the following section.

3.2 Benefit Function: The Utility of Risk Reduction

Consider designing a single subsystem (e.g., propulsion system) of n elements where risk reduction at each element may contribute to the health of other elements. By allocating resources during the design stage to mitigate the risk of each functional element, one can reduce the risk of the overall subsystem as a whole. We quantify the benefit (i.e. utility) of allocating resources to reduce design risks at each element as the amount of risk reduction– referred to as *Risk Benefit Function*, or in short *Benefit Function* in the rest of this paper (denoted by b_i). Note that early in the design process, the final benefit (i.e. risk reduction) that will be achieved by consuming one unit of risk-mitigation resources is not known. Therefore, b_i is treated as a stochastic process in RUBIC.

Definition: The *benefit function* of spending 1 unit of resource to reduce risk at the i-th element, denoted by b_i, is a random process with a given mean and variance. The expected outcome of spending 1 unit of such resource is $\mu_i = E(b_i)$[2], which represents the expected risk reduction.

To simplify the mathematical process, we assume a certain form of probability distribution function for this stochastic process, and relate mean and variance in order to reduce its degrees of freedom to 1 (Later we will show that the optimal resource allocation vector in

[2] In this paper, E(.) and Var (.) refer to the expected value and variance of a random process, respectively.

RUBIC is scaling independent and therefore, the scaling of the *pdf* does not affect the outcome). Here we assumed a benefit function with a triangular distribution (see Figure 2.)

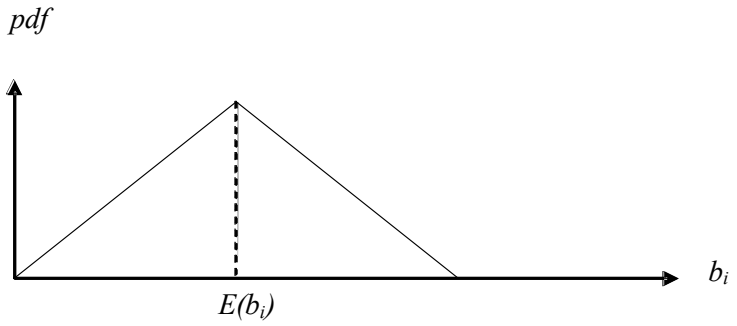

Figure 2: Triangular distribution for the random process, b_i

μ_i can be estimated from historical data or theoretical models (more on this later in this paper). Assuming a triangular distribution of the above form[3]:

$$\sigma_i \approx 0.3\ \mu_i \tag{1}$$

It will be argued later in this paper that the constant factor (that is derived from the triangular distribution assumption) will not affect the outcome of RUBIC (the approach is scaling independent). In some instances, however, we maybe interested in absolute measures of risk reduction in which case, such assumptions are necessary.

3.3 Single-Subsystem RUBIC Design Methodology

Assume that a system design project has a certain amount of risk reduction resources that can be distributed in a single-subsystem of *n* elements. The questions that need to be answered are: Which risk factors are the most crucial ones? How must the resources be allocated? The answer to these questions can have a significant impact on the performance and effectiveness of the risk reduction process. RUBIC guides the design process, given the available risk reduction resources, such that the end design is minimally susceptible to failure. The allocation of these resources among functional elements is referred to as *risk reduction resource allocation vector,* or *allocation vector* for short:

[3] In this paper, σ_{ii} denotes Var(b_i); σ_{ij} refers to Cov(b_i,b_j), and σ_i (with one index) refers to the standard deviation of b_i, i.e. $\sigma_{ii} = \sigma_i^2$

Definition: *Risk reduction resource allocation vector*, denoted by $\mathbf{w} = [w_1, \ldots, w_n]^T$, is defined as the percentages of resources to be spent on the n functional elements within a system.

The goal of the RUBIC design methodology is to determine **w** dynamically throughout the design process. Based on this evolving allocation vector, designers can sort their priorities and allocate optimal amount of resources (e.g., time or money) to reduce risk of each functional element. This will lead to the concept of "Risk-Efficient Design Process" or RED-P, as described later in this section for a single-subsystem case.

Given the allocation vector, **w**, we can compute the total benefit function as: $TB = \sum_{1}^{n} w_i b_i$. Since b_i is a random process (that is unknown during the design process), we would like to:

1. Maximize the expected total benefit, i.e., $E(TB) = \mathbf{w}^T\boldsymbol{\mu}$
2. Minimize the variance of total benefit, i.e., $Var(TB)=\mathbf{w}^T\boldsymbol{\Sigma}\mathbf{w}$

where $\boldsymbol{\mu}$ is the vector of expected values for b_i's, and $\boldsymbol{\Sigma}$ is the covariance matrix (a diagonal element, σ_{ii} , is the variance of b_i, while an off-diagonal element, σ_{ij}, reflects the covariance of risk in elements i and j), i.e.,

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \ddots \\ \mu_n \end{bmatrix} \begin{matrix} \leftarrow \text{element 1} \\ \\ \leftarrow \text{element } n \end{matrix}$$

and;

$$\Sigma = \begin{bmatrix} \sigma_{1\,\text{element 1}} & \cdots & \sigma_{1\,\text{element } n} \\ \vdots & \ddots & \vdots \\ \sigma_{n\,\text{element 1}} & \cdots & \sigma_{n\,\text{element } n} \end{bmatrix}$$

element 1

element n

Note the resemblance of the above formulation to Markowitz's optimal portfolio selection problem (that is based on relatively similar assumptions in the context of managing risky financial assets. See Markowitz 1952). This formulation of risk reduction is in fact a two-objective optimization problem:

$$\begin{cases} \text{Minimize } \mathbf{w}^T\Sigma\mathbf{w} \\ \text{Maximize } \mathbf{w}^T\boldsymbol{\mu} \\ \qquad \in F \\ \text{s.t. } \mathbf{w} \end{cases} \tag{2}$$

where F is the set of all feasible design processes (within design constraints as well as discretionary limitations set forth by the design team). This bi-criterion optimization problem results in a set of Pareto optimal solutions (also referred to as an efficient frontier) that outlines the optimal tradeoff between safeguarding only the most risky elements (with highest risk premiums) versus trying to diversify risk-reduction throughout the system (two different extremes). Later in the example of this paper, we use a simple method to account for this tradeoff based on preferences. We refer to a solution to the above bi-criterion optimization problem as a *Risk-Efficient Design Process* or RED-P.

Definition: A *Risk-Efficient Design Process* or RED-P is one that is optimal with respect to Equation 2.

Figure 3 shows the set of all feasible design processes and the subset of risk-efficient ones (identified by a thick curve). Every point on the efficient frontier is considered a RED-P. Other design processes that fall inside the feasible domain are inferior with respect to the points on the RED-P curve because they offer a higher variance and a lower expected benefit (worse with respect to both criteria).

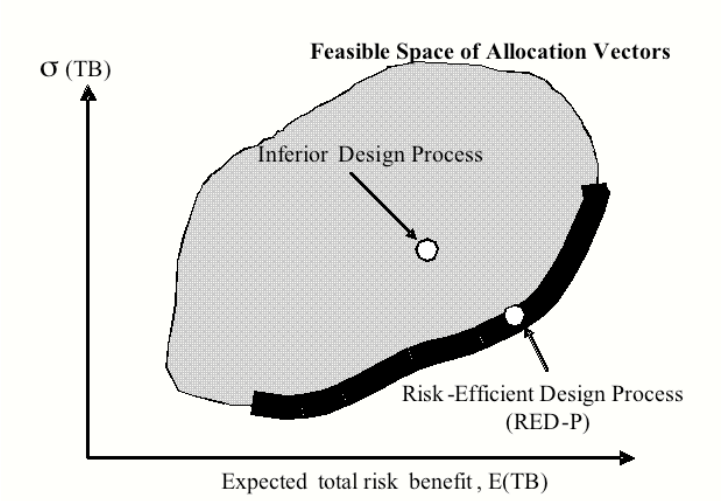

Figure 3: The Thick curve represents the efficient frontier (set of all feasible RED-P's)

In the RUBIC design methodology, a design process that lies on the efficient frontier is considered acceptable while all others are considered unacceptable (because they represent an inefficient use of risk-mitigation resources). By solving Equation 2 and choosing a RED-P on the efficient frontier, we obtain an allocation vector, **w**, which can be used to rank order areas of focus for reducing risk and the amount of resources to be used on each element. This will provide the decision-makers with guidelines to improve the reliability of the subsystem in an efficient way. This is demonstrated in the following section for the case of a single subsystem design using the reaction wheel example of Figure 1.

3.4. Example: Single-Subsystem RUBIC Design Method for a Satellite Reaction Wheel

In this section, we will only focus on the Motor Controller subsystem of Figure 1 (Later in this paper, we will return to this problem from a multi-subsystem approach). In the Motor Controller subsystem, there are 7 functional elements, as listed in the following:

- Import Electrical Energy
- Export Electrical Energy
- Guide Electrical Energy
- Regulate Electrical Energy
- Guide Electrical Energy
- Condition Electrical Energy
- Guide Electrical Energy

Note that each of these function elements may correspond to one or more physical element in the subsystem. Some of these functional elements, such as 'Regulate E. Energy', correspond to a complex circuitry while some others, such as 'Guide E. Energy', correspond to a simple physical element such as a wire. Since this functional model corresponds to a preliminary design and has not yet converged to a detailed design, these functional elements are relatively generic. As discussed before, this is where a thorough risk analysis and optimal resource allocation can have the greatest impact on the overall safety of the final design. There is significant amount of historical failure data from which $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ can be estimated at this stage using FFD and other similar methods (the estimation methods will be discussed in Section 4). $\boldsymbol{\mu}$ is proportional to the failure rate (expected individual risk premium of each functional element). Note that RUBIC is scaling independent since a constant scale can be factored out in the optimization problem of Equation 2. Therefore, the absolute values in $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ do not affect the optimal allocation vector. From Equation 1, we can also estimate σ_{ii}'s from μ_i's (Again, note that the constant factor can be factored out). σ_{ij}'s are also estimated from incidents where a malfunction in one functional element led to failure in another element (thereby capturing interactions.) The following is a summary of numerical estimations (constant multipliers are factored out and not shown):

$$\boldsymbol{\mu} = \begin{bmatrix} 0.03 \\ 0.03 \\ 0.05 \\ 0.45 \\ 0.05 \\ 0.33 \\ 0.05 \end{bmatrix} \begin{matrix} \text{Import Elec. E.} \\ \text{Export Elec. E.} \\ \text{Guide Elec. E.} \\ \text{Regulate Elec. E.} \\ \text{Guide Elec. E.} \\ \text{Condition Elec. E.} \\ \text{Guide Elec. E.} \end{matrix}$$

and;

$$\Sigma = \begin{bmatrix} 9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 9 & 0 & 1021 & 0 & 0 & 0 \\ 0 & 0 & 25 & 0 & 0 & 0 & 0 \\ 0 & 1021 & 0 & 2025 & 81 & 215 & 0 \\ 0 & 0 & 0 & 81 & 25 & 81 & 0 \\ 0 & 0 & 0 & 215 & 81 & 1089 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 25 \end{bmatrix} \times 10^{-4}$$

Using the above numerical values, one can solve Equation 2 to obtain the risk-efficient design frontier. In Figure 4, a sample space is plotted by choosing random values for the

allocation vector (i.e., **w**) and plotting the mean and standard deviation values. This figure also shows the approximate location of the efficient frontier. As explained before, every point on this frontier is a Risk-Efficient Design Process (which implies that risk reduction resource is optimally allocated among functional elements). Every design process that is not located on this frontier is inefficient in the sense that a higher expected value and a lower variance can be obtained for the total risk benefit function by re-allocating resources (which implies that the resource allocation vector is not optimal).

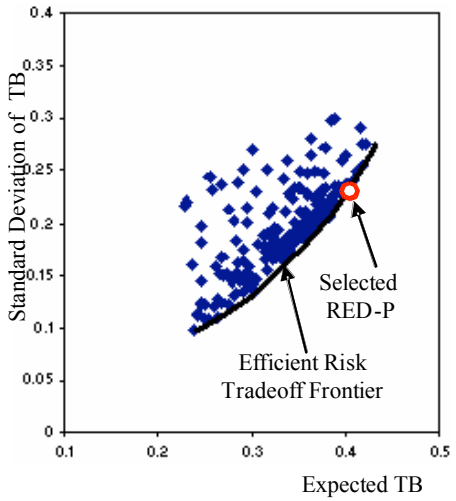

Figure 4: The efficient risk tradeoff frontier

Now the question becomes which RED-P on the efficient frontier should be chosen. In fact, the efficient frontier represents the tradeoff between expected value and variance of the total benefit function. On one extreme, the tradeoff tends to maximize expected value (by focusing on the most risky elements only). However, focusing solely on the most risky elements would ignore other elements in the system (which might cause failure, particularly due to high covariance). The other extreme of the tradeoff tends to diversify the resources to minimize the variance of the observed benefit. To keep the problem simple, we use a linearly weighted utility function to assess the tradeoff between these two criteria[4]: $u=E(TB)-0.3\sigma(TB)$. The negative sign of $\sigma(TB)$ accounts for the fact that it needs to be minimized. Note that in practice, these weights can be obtained from the designers involved in the process. One may even choose to vary these weights dynamically throughout the design process. For instance, one may decide to assign a

higher weight to E(TB) early in the design process (to gain as much risk reduction as possible early in the design process). However, as the design converges closer to the final design, one may choose to increase the weight for σ(TB) to better spread the risk reduction resources. Using the above linear utility, the two-objective optimization problem of Equation 2 collapses to that of maximizing utility. This will result in a single allocation vector that corresponds to the most preferred RED-P, identified by a red circle in Figure 4 and listed in Table 1.

Table 1: Optimal resource allocation (corresponds to the red circle in Figure 4)

Column #	Function	**Resource Allocation**
1st	Import Electrical Energy	**<1%**
2nd	Export Electrical Energy	**6%**
3rd	Guide Electrical Energy	**<1%**
4th	Regulate Electrical Energy	**57%**
5th	Guide Electrical Energy	**10%**
6th	Condition Electrical Energy	**26%**
7th	Guide Electrical Energy	**<1%**

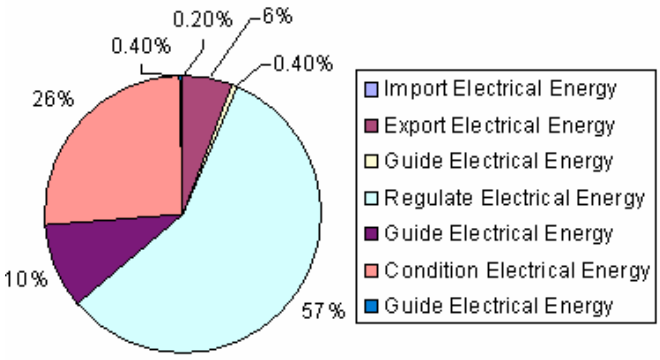

Figure 5 – Optimal Allocation

From Figure 5, it is clear that two functional elements (i.e. 'Regulate Elec. E.' and 'Condition Elec. E.) require the highest resources. Another interesting observation is that the 5th functional element which has the same functionality as the 3rd and 7th elements (i.e. they 'Guide Elec E.') has assumed a relatively higher priority than the other two. This is mainly because the high covariance values between this particular electrical connection with 4th and 6th elements (this is basically the electrical bridge between these two high-risk components). Also, the function 'Export Elec. E.' has assumed a relatively high value despite the fact that it has the

[4] Note that we chose σ(TB) instead of Var(TB) because σ(TB) and E(TB) have the same units and can be used

lowest failure rate (because of its correlation with failure in the 4th element; See the high corresponding covariance value in the covariance matrix). In fact, this can be related to the fact that failure in 'exporting electrical energy' (which conducts static charges out of the circuitry to the body of satellite) may cause failures due to the accumulation of static electricity in the 'regulate' and 'condition' functions. This is an example of an observation that is hard to make without using the RUBIC design method. In the following section, RUBIC is extended to the case of multi-subsystem designs.

3.5 Multi-Subsystem RUBIC Design Methodology

Consider the case of a system of m subsystems, denoted by $S_1 \ldots S_m$, containing $n_1 \ldots n_m$ elements, respectively (n_i>0). In Section 3.1, we stated that the risk of two elements in the same subsystem may be correlated, while the risks of elements in different subsystems are independent, i.e.,

$$\sigma_{i,j-k,l} = 0 \qquad \text{if } i \neq k;\ 1 \leq i \leq n_i,\ 1 \leq k \leq n_k$$

where $\sigma_{i,j-k,l}$ refers to the covariance of risk between the j-th element in the i-th subsystem and the l-th element in the k-th subsystem. The RUBIC design methodology can then be simply extended to the multi-subsystem case, as in the following:

and;

$$\boldsymbol{\mu} = \begin{bmatrix} \begin{bmatrix} \mu_{11} \\ \\ \mu_{1n_1} \end{bmatrix} \\ \\ \\ \begin{bmatrix} \mu_{m1} \\ \\ \mu_{mn_m} \end{bmatrix} \end{bmatrix}$$

1st subsystem

m-th subsystem

in a linear combination.

$$\Sigma = \begin{bmatrix} \begin{bmatrix} \sigma_{11-11} & \sigma_{11-1n_1} \\ \sigma_{1n_1-11} & \sigma_{1n_1-1n_1} \end{bmatrix} & & 0 & 0 & 0 \\ & \overbrace{\quad}^{1^{st}\text{ subsystem}} & & & \overbrace{\quad}^{m\text{-th subsystem}} \\ 0 & & \begin{bmatrix} & \\ & \end{bmatrix} & & 0 \\ 0 & & & & \\ 0 & 0 & 0 & \begin{bmatrix} \sigma_{m1-m1} & \sigma_{m1-mn_m} \\ \sigma_{mn_m-m1} & \sigma_{mn_m-mn_m} \end{bmatrix} & \end{bmatrix}$$

A risk-efficient design process can then be obtained in a similar fashion to the single-disciplinary case by solving Equation 2. Note that the problem in the multi-subsystem case is to allocate resources among subsystems and then among functional elements within each subsystem. This is demonstrated in the following example.

3.6 Example: Multi-Subsystem RUBIC Design for a Satellite Reaction Wheel

In this section, the RUBIC design methodology is used to analyze risk in all subsystems of the satellite reaction wheel (Figure 1). The system consists of 4 subsystems:

- Motor Controller Subsystem (7 functional elements)
- Motor Subsystem (3 functional elements)
- Flywheel Subsystem (3 functional elements)
- Structure Subsystem (5 functional elements)

In a similar fashion to the single-subsystem case, the multi-subsystem analysis starts with estimations for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ (more on estimations methods in Section 4):

$$\cdot = \begin{bmatrix} 0.03 \\ 0.03 \\ 0.05 \\ 0.45 \\ 0.05 \\ 0.33 \\ 0.05 \\ 0.16 \\ 0.21 \\ 0.06 \\ 0.05 \\ 0.00 \\ 0.08 \\ 0.00 \\ 0.00 \\ 0.05 \\ 0.00 \\ 0.00 \end{bmatrix} \begin{matrix} \\ \text{Import Elec. E.} \\ \text{Export Elec. E.} \\ \text{Guide Elec. E.} \\ \text{Regulate Elec. E.} \\ \text{Guide Elec. E.} \\ \text{Condition Elec. E.} \\ \text{Guide Elec. E.} \\ \text{Convert Elec. E. to Rot. E.} \\ \text{Convert Rot. E. to Elec. E.} \\ \text{Export Therm. E.} \\ \text{Transfer Rot. E.} \\ \text{Store Rot. E.} \\ \text{Supply Rot. E.} \\ \text{Import Solid} \\ \text{Secure Solid} \\ \text{Transfer Rot. E.} \\ \text{Export Solid} \\ \text{Export Rot. E.} \end{matrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} \\ \\ \\ \\ \text{Motor Controller Subsystem} \\ \\ \\ \\ \text{Motor Subsystem} \\ \\ \\ \text{Flywheel Subsystem} \\ \\ \\ \text{Structure Subsystem} \\ \\ \\ \\ \end{matrix}$$

and;

$$\Sigma = \begin{bmatrix} \begin{bmatrix} 9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 9 & 0 & 1021 & 0 & 0 & 0 \\ 0 & 0 & 25 & 0 & 0 & 0 & 0 \\ 0 & 1021 & 0 & 2025 & 81 & 215 & 0 \\ 0 & 0 & 0 & 81 & 25 & 81 & 0 \\ 0 & 0 & 0 & 215 & 81 & 1089 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 25 \end{bmatrix} & 0 & 0 & 0 \\ 0 & \begin{bmatrix} 256 & 0 & 112 \\ 0 & 441 & 982 \\ 112 & 982 & 36 \end{bmatrix} & 0 & 0 \\ 0 & 0 & \begin{bmatrix} 25 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 64 \end{bmatrix} & 0 \\ 0 & 0 & 0 & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 25 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{bmatrix} \times 10^{-4}$$

Figure 6 depicts a random sample of feasible design processes as well as the approximate location of the risk-efficient frontier. Using the utility function of Section 3.4, we obtain the

desired RED-P (indicated by a red circle). The corresponding optimal allocation vector is listed in Table 2.

Table 2: Optimal resource allocation (red circle in Fig 6)

Column #	Subsystem	Function	**Resource Allocation**
1st	Motor Controller	Import Electrical Energy	**<<1%**
2nd	Motor Controller	Export Electrical Energy	**4%**
3rd	Motor Controller	Guide Electrical Energy	**<<1%**
4th	Motor Controller	Regulate Electrical Energy	**36%**
5th	Motor Controller	Guide Electrical Energy	**6%**
6th	Motor Controller	Condition Electrical Energy	**17%**
7th	Motor Controller	Guide Electrical Energy	**<<1%**
Total Allocation to Controller Subsystem: 64%			
8th	Motor	Convert Electrical E. to Rotational E.	**9%**
2nd	Motor	Convert Rotational E. to Electrical E.	**13%**
3rd	Motor	Export Thermal Energy	**10%**
Total Allocation to Motor Subsystem: 32%			
1st	Flywheel	Transfer Rotational Energy	**1%**
2nd	Flywheel	Store Rotational Energy	**0%**
3rd	Flywheel	Supply Rotational Energy	**2%**
Total Allocation to Flywheel Subsystem: 3%			
	Structure	Import Solid	**0%**
	Structure	Secure Solid	**0%**
	Structure	Transfer Rotational Energy	**1%**
	Structure	Export Solid	**0%**
	Structure	Export Rotational Energy	**0%**
Total Allocation to Structure Subsystem: 1%			

The controller and motor subsystems pose the highest risk premiums to the overall health of the system. Note that, as the design process evolves, the functional model evolves and new failure modes may appear. The main advantage of RUBIC is that it can determine the optimal allocation of resources in real-time, i.e., as new functional elements appear (or the old ones are removed, modified, or decomposed to more functional elements) the optimal allocation vector

adjusts accordingly to identify the critical areas of design that pose maximum risk to the overall health of the system.

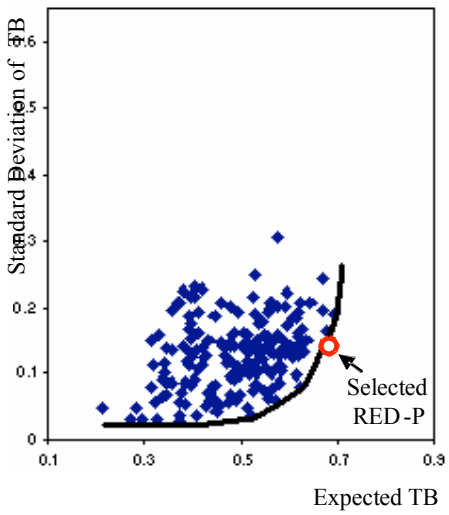

Figure 6: The efficient risk tradeoff frontier

4. ESTIMATION OF FUNCTIONAL FAILURE RATES

There are a variety of techniques in the literature for extracting failure occurrence rates from historical data or expert opinion[5]. We propose using a derivative of Function Failure Design (FFD) method (Tumer and Stone, 2003) that obtains actual failure rates based on the composite function failure matrix. FFD is based on the premise that failure modes can be traced back to the functions that a particular component addresses. The functional model can therefore be mapped to failure modes via the FFD method. The obtained failure rate estimations can then be used in RUBIC to make objective and numerically verifiable decisions during the design process (Refer to Stone et al. 2005 and Stock et al. 2005, for a detailed review of this method and how it can be used to obtain dynamic failure analysis information). In short, FFD has 5 major steps:

1- Document Functional Data: The first step is to develop a functional model for the system

2- Create a function-component matrix: The components form the *m* columns of this matrix and the functions form the *n* rows. For a given component a '1' is placed in the cell

[5] FMEA for instance, assigns a value to the failure rate based on reasonable estimations of the probability of occurrence obtained from experienced designers.

corresponding to the function it performs and a '0' is placed in other cells. This is referred to as the EC matrix which correlates physical components of a system with the functional model.

3- Document the Failure Data: This step involves obtaining feedback information (from historical data or expert elicitation) about all potential failure modes and their causes.

4- Create Component Failure Mode Matrix: This matrix has *p* columns representing the failure modes and *n* rows representing the components and is denoted by **CF**. As in the function-component matrix, a '1' in this matrix represents a component that has experienced a certain failure mode.

5- Obtain Function-Failure Matrix: This is obtained by multiplying the function-component matrix and the component-failure mode matrix, i.e., **EF = EC x CF.** This matrix represents the number of occurrences of a particular failure mode for a given function, from which occurrence ranking values could be obtained using the probability of occurrence. The probability could be obtained from the ratio of the number of occurrences of a failure to the total number of instances of failure.

Using FFD requires building a large and comprehensive knowledge base of failure modes and their occurrences. Once this knowledge base is developed, it can be used to obtain numerical estimations in real-time. Using FFD allows the RUBIC design methodology to evaluate optimal allocation vector dynamically as the design progresses (for example, via a network-based repository in a real-time fashion.). An example of such real-time network-based tool is currently under development as part of an ongoing project at NASA Ames Research Center. Upon completion, a RUBIC design tool will be able to perform function-based queries to this knowledge base to retrieve failure estimations using the FFD method.

5. APPLICATION OF RUBIC DESIGN METHODOLOGY TO THE DESIGN OF SPACE EXPLORATION VEHICLES AT NASA

Among various aerospace systems, the design of NASA's space exploration vehicles is particularly risk-driven due to the high-cost, low-volume, and social and scientific impacts of such space missions. While these complex engineering system are susceptible to failure and ultimately loss of mission, the current risk management practice is mostly ad-hoc and based on asking "what can go wrong?" from the experts. The importance of finding a rigor for risk-based

design decision making is also boosted by the fact that NASA uses concurrent engineering teams to reduce the time and costs associated with the early design study of space missions. There are several real time concurrent design teams at various NASA centers. For example, Team X at the Advanced Project Design Center at NASA's Jet Propulsion Laboratory is a concurrent engineering team that produces conceptual designs of space missions for the purpose of analyzing the feasibility and estimating the cost of mission ideas proposed by its customers. The study takes one to two weeks and the final design is then documented in a 30 to 80-page report that includes equipment lists, mass and power budgets, system and subsystem descriptions, and a projected mission cost estimate. A design decision that is made during this phase often has a significant impact on the overall cost and success of a mission as well as its associated risk and uncertainty. As mentioned before, current practice does not pay adequate attention to capturing and describing the risk elements associated with the final design. It is often unclear from the final report that why certain design decisions were made, what options were considered, and what was the potential risk tradeoff between these options. Due to the lack of information about the rationale involved in making these decisions, it is often very difficult to verify such decisions and their role in the overall safety of the mission. Most of current efforts use qualitative techniques that would list potential failures based on inputs from engineers, i.e., expert-elicitation techniques. (See for instance Meshkat and Cornford 2003). These techniques, however, have several shortcomings that can limit their effectiveness – prohibiting a thorough study of failure elements and probabilities in such dynamic design environments:

- Due to the numerous dependencies that exist between the various subsystems in a spacecraft and the speed with which the engineers make design decisions, the subsystem engineers are sometimes unaware of the important design choices of others. Since each design option correlates with particular types of risks, the only way to keep the engineers informed about the design options under consideration is by informing all of them about all risk elements related to them dynamically (i.e., live information feed). This becomes increasingly difficult as the complexity of the design grows. For distributed design teams, in particular, the back and forth communication of this huge amount of data between all subsystems is impractical.

- A major shortcoming of approaches that are currently used in concurrent design environment is that they do not provide solid quantitative risk measures to guide the engineers in the decision-making process. These decisions include selecting among alternative designs while trading-off

risk with other objectives such as weight, cost and performance. This is a particular problem for NASA because design constraints (e.g., cost and weight limits) are fairly rigid and the feasible design space is extremely tight.

- Another major challenge lies in the integration of these risk analysis methods with such rapidly evolving design processes. Many of these methods require a fully converged design. So they integrate well into a system design process after major review stages, but these same methods cannot be applied during earlier phases when tenuous design decisions are made and withdrawn rapidly.

The proposed approach of this paper helps improve at least some aspects of such design efforts, as described in the following.

- The RUBIC design methodology provides a rigorous quantitative framework for considering risk and uncertainty during conceptual design of space exploration systems (which is the focus of these rapid concurrent engineering teams).

- RUBIC assumes a hierarchical decomposition of a system (Concurrent engineering teams almost always view a design problem in the same manner by decomposing it into subsystems)

- RUBIC is based on functional modeling of a system. So, as the design process evolves, its functional model evolves. This allows for an easy integration of the RUBIC design methodology with the evolution of the design and makes it applicable to all stages of this process (from earlier conceptual design to preparing the risk report for the final design).

- RUBIC provides the capability to compute allocation decisions in a real-time fashion. This allows the system-level decision-makers to dynamically adjust to the design decisions as the overall system evolves throughout the design process.

6. EPILOGUE

In this paper, we introduced RUBIC as a risk-driven methodology that can be used during the concurrent design of aerospace systems. The objective is to reduce risk in various subsystems and functional elements given the available risk-mitigation resources. The proposed approach is based on the notion that a failure happens when a functional element in the system does not perform the intended task. One unit of risk-mitigation resource then will reduce the risk of that certain functional-failure happening and therefore, contribute to the overall system risk

reduction. The resource allocation problem can then be formulated as a two-objective optimization problem. We defined a Risk-Efficient Design Process (or RED-P) as one that is optimal with regard to this optimization problem. The proposed approach was demonstrated using a satellite reaction wheel design problem. The RUBIC design methodology was then used to leverage a knowledge base of failure data to identify potential risks during the design of such system. It was showed through this example that without using the RUBIC design method, it is often very difficult to make numerically verifiable risk reduction decision during the conceptual design of complex multi-level systems. In particular, in a concurrent and distributed design environment where design decisions are made and withdrawn very quickly, only numerical and real-time methods such as RUBIC are capable of providing an insight into major contributing risk factors and their propagation in the system. An ongoing project at NASA Ames Research Center is developing a network-enabled failure knowledge base that will be used to support the RUBIC design tool. Upon completion, this tool will be able to provide real-time risk guidance to concurrent and distributed engineering design teams throughout the design cycle of space exploration missions.

REFERENCES

Backman, B., 2000, “Design Innovation and Risk Management: A Structural Designer's Voyage into Uncertainty,” ICASE Series on Risk-based Design, November 2000.

Choi, K., 2001 “Advances in Reliability-Based Design Optimization and Probability Analysis - PART II”, ICASE Series on Risk-based Design, December 2001.

Du, X., Chen, W., 2002, “Efficient Uncertainty Analysis Methods for Multidisciplinary Robust Design”, AIAA Journal, 40(3), 545-552.

Greenfield, M. A., 2000, “NASA's Use of Quantitative Risk Assessment for Safety Upgrades”, Proceedings of the IAA Symposium, Rio de Janeiro, Brazil.

Hunt, J. E., Pugh, D. R. Price, C. P., 1995, “Failure Mode Effects Analysis: A Practical Application of Functional Modeling,” Applied Artificial Intelligence, Vol. 9(1), pp33-44.

Markowitz, H., 1952, “Portfolio Selection”, Journal of Finance, Vol. 7(1), pp. 77-91.

Meshkat, L., Cornford, S., “Risk Based Decision Tool for Space Exploration Missions”, Proceedings of the AIAA Space Conference, 2003.

Rose, J., “Risk Management for Jet Propulsion Laboratory Project”, ASME/SERAD International Mechanical Engineering Congress and Exposition, Orlando, Florida, 2000.

Schrader, S., Riggs, W., Smith, R.P., 1993, “Choice over Uncertainty and Ambiguity in Technical Problem Solving,” Journal of Eng. & Technology Mgmt, Vol. 10, pp. 73-99.

Smith, N., Mahadevan, S., “Probabilistic Methods for Aerospace System Conceptual Design,” Journal of Spacecraft and Rockets, AIAA, Vol. 40, No. 3, pp. 411-418, 2003.

Stock, M.E., Stone, R.B., Tumer, I. Y., “Linking product function to historical failures to improve failure analysis in design” Research in Engineering Design. In Print. 2005.

Stone, R.B., Wood, K.L, 2000, “Development of a Functional Basis for Design”, Journal of Mechanical Design, Vol. 122, pp. 359-370.

Stone, R., Wood, K., Crawford, R., 2000, “Using quantitative functional models to develop product architectures”, Design Studies, Vol. 21(3), pp. 239–260.

Tumer, I. Y., Stone, R.B., “Mapping Function to Failure During High-Risk Component Development” Journal of Research in Engineering Design, Vol. 14, pp.25-33. 2003.

Stone, R.B., Tumer, I.Y., VanWie, M. “The function-failure design method.” Journal of Mechanical Design. 2005.

Zang, T. A., Hemsch, M. J., Hilburger, M.W., Kenny, S. P., Luckring, J. M., Maghami, P., Padula, S. L., Stroud W. J., 2002 “Needs and Opportunities for Risk-Based Multidisciplinary Design Technologies for Vehicles”, NASA TM, July 2002.