Quantifying the Resilience-Informed Scenario Cost Sum (RISCS): A Value-Driven Design Approach for Functional Hazard Assessment

Daniel Hulse
Graduate Research Assistant
School of Mechanical, Industrial and Manufacturing Engineering
Oregon State University
Corvallis, Oregon 97330

Christopher Hoyle *
Associate Professor
School of Mechanical, Industrial and Manufacturing Engineering
Oregon State University
Corvallis, Oregon 97330

Kai Goebel
Tech Area Lead
Discovery and Systems Health
Intelligent Systems Division
NASA Ames Research Center
Moffett Field, California 94035
Adjunct Professor
Luleå Technical University
Division of Operation and Maintenance Engineering
Luleå, Sweden

Irem Y. Tumer
Professor
School of Mechanical, Industrial and Manufacturing Engineering
Oregon State University
Corvallis, Oregon 97330

ABSTRACT

Complex engineered systems can carry risk of high failure consequences, and as a result, resilience–the ability to avoid or quickly recover from faults–is desirable. Ideally, resilience should be designed-in as early in the design process as possible, so that designers can best leverage the ability to explore the design space. Towards this end, previous work has developed functional modeling languages which represent the functions which must be performed by a system and function-based fault modeling frameworks have been developed to predict the resulting fault propagation behavior of a given functional model. However, little has been done to formally optimize or compare designs based on these predictions, partially because the effects of these models have not been quantified into an objective function to optimize. The work described herein closes this gap by introducing the resilience-informed scenario cost sum (RISCS), a scoring function which integrates with a fault scenario-based simulation, to enable the optimization and evaluation of functional model resilience. The scoring function accomplishes this by quantifying the expected cost of a design's fault response using probability information, and combining this cost with design and operational costs such that it may be parameterized in terms of designer-specified resilient features. The usefulness and limitations of using this approach in a general optimization and concept selection framework are discussed in general, and demonstrated on a monopropellant system design problem. Using RISCS as an objective for optimization, the algorithm selects the set of resilient features which provides the optimal trade-off between design cost and risk. For concept selection, RISCS is used to judge whether resilient concept variants justify their design costs and make direct comparisons between different model structures.

1 Introduction

Complex engineered systems such as nuclear power plants, aerospace vehicles, and oil rigs are often associated with large investments and significant failure consequences. High-profile failures in these systems, such as the Chernobyl disaster

*Address all correspondence to this author.

[1], Challenger catastrophe [2], and Deepwater Horizon oil spill [3] have caused deaths, environmental damage, and billions of dollars of economic loss. It is therefore desirable to design the systems in such a way that minimizes risk and responds well to adverse circumstances so that performance, cost, and safety are maintained or recovered.

This goal presents a significant challenge, due to the inherent complexity of these systems. Indeed, complex engineered systems comprise many components, each with many possible interactions. Despite each component having relatively low failure probability, there is often only a poor understanding of compound failure risk. For example, in the aftermath of the Challenger catastrophe, it was found that engineers' estimates of overall failure probability differed by orders of magnitude [2, see Appendix F]. While failures in complex systems are often attributed to poor management and operations, they can often be traced to design flaws.

To address this challenge, risk and failure approaches have been introduced which help designers reason about failures and their impact on the design [4], including failure modes and effects analysis [5], fault tree analysis [6], and model-based approaches [7] [8]. However, these approaches do not generally consider resilience–the ability of the system to recover from failures–since they are based on assessing failure only. Furthermore, these approaches are generally not suitable for early design–the focus of this work–since they are based on detailed knowledge a fully realized system. To evaluate a design's failure risk without being locked-in to a specific realization, failure approaches have been formulated based on the functional model of the system, since these are available earlier in the design process.

For preliminary and conceptual design, Functional Hazard Assessment has been recommended within the aerospace industry as a method to enable designers to proactively identify hazards which happen at the functional level so that they can be eliminated or mitigated by design [9] [10] [11]. Functional Hazard Assessment involves the systematic identification of hazards associated with system functions and interactions [12] [10], and may be aided by building a fault model based on the functional model of the system to capture the propagation of errors [13], or by building a dynamic model of how the system interacts with its environment [14]. This model may then be further developed and added to through the rest of the system assessment and design process [15]. While functional hazard assessment is a process to identify risks, it is not in and of itself a design method, but instead simply informs the design process about relevant hazards in the system.

1.1 Prior Work

To incorporate functional hazard information in design, prior work introduced formal methodologies to enable designers' use and understanding of the information. The function-failure design method (FFDM) to predict likely failure modes due to the loss of functions using past data to show which functions require more design attention [16]. This was extended in the risk in early design method (RED) using likelihood and consequence estimates to better inform designers [17] [18], which has since been shown to better allow students to assess risk [19]. To analyze social and organizational hazards in engineering systems, the Functional Resonance Analysis Method (FRAM) was developed to analyze hazards within socio-technical systems based on high-level functional relationships [20], such as air traffic management [21]. This socio-technical interaction has also been analyzed using a hierarchical functional decomposition of a system to identify hazards within process plants [22] and food processing [23]. These tools are based on building tables or databases of previous faults which occurred within functions, as well as their effects concurrently with a functional model.

Other approaches use function-based computational models to encode risks. Hierarchically Performed Hazard Origin and Propagation Studies (HiP-HOPS) was developed as a tool to assess risk throughout a hierarchical system model by integrating functional and classical techniques into a single consistent model [24]. Rather than modeling failures within engineered systems as a result of component failures, the Systems-Theoretic Accident Model and Processes (STAMP) models the dynamics of the organizational environment to find and redesign inadequate control processes that lead to failure [25] [26] [27] [28]. The function failure identification and propagation method (FFIP) informs assessment by constructing a graph-based behavioral model to take into account the function interactions, dynamics, and joint fault scenarios [8] which has since been extended using flow-state logic to model undesired flow states [29] and dimensional analysis to incorporate more detailed information about component behavior [30] and adapted for large-scale complex systems [31] and mechatronics [32]. Inherent Behavior of Functional Models (IBFM), which is used in this paper, provided a method to automate the creation of a state-based behavior model from the functional model itself [33]. Approaches have additionally been presented which associate the functional model with a fault tree [34] [35] [36], and other methods have been created to focus on the propagation of failures through a functional model [37].

Attempts have in turn been made to show how to generate, improve, or change the design based on these function-based failure frameworks. Initially in developing these frameworks, the resulting information was simply used to show designers where attention should be paid in making design choices [16]. Approaches have subsequently been presented to use graph grammars to change the structure of the model, and/or use a cost-risk analysis scoring function to compare between design alternatives [38] [39]. Additionally, an approach has been presented for designing the operational decision-making in the model to determine when to, for example, route degraded flows to sacrificial subsystems [40]. While these approaches show many of the design changes that can be made within a functional model, and can be used to compare between design alternatives, they do not use this knowledge to formally optimize a design problem.

A few approaches for formal optimization of risk within functional models have additionally been presented. Using the HiP-HOPS risk modelling methodology, design optimization [41] and optimization of fault tolerance [42] has been demonstrated. Additionally, using functional-failure-matrix approaches, the Risk and Uncertainty Based Concurrent Integrated Design Methodology (RUBIC) first introduced the concept of using failure scenarios, probabilities, and costs to optimize risk within a functional model to show where to allocate resources based on function-failure matrices used in FFDM [43]. Additionally, objectives been formed for the allocation of health management within a functional model informed by the effect of adding sensors, reducing failure probabilities, and changing inspection intervals on reducing overall design risk [44]. While these approaches show that some optimization has been performed in the context of function-failure methodologies, no approaches have yet been presented for the graph-based fault models, such as IBFM and FFIP.

1.2 Aims and Contributions

The objective of this research is to create a framework for concept selection and optimization to approach resilient design early using function-based fault models. Towards that goal, this paper presents the Resilience-Informed Scenario Cost Sum (RISCS), a scoring function which integrates with a model-generated set of fault-event simulations to calculate the expected cost of a design considering every fault scenario. This scoring function combines this expected cost of risk with the design and operational costs to resolve the trade-offs inherent between cost, risk, and performance.

As a single, holistic design measure, this scoring can then be used as the sole value consideration both to guide early design processes and to be given as an objective in optimization procedures. By using this measure, both processes can make proactive, risk-informed design decisions at the earliest stages of design, while there is the most ability to meaningfully impact design resilience. Using this scoring in typical systematic design process (such as that described in [45, see Chapter 2]), designers can compare a variety of early functional design concepts and determine how best to make key design decisions, such as function structure, high-level functional requirements, and solution working principles. Using the scoring for optimization, solution procedures may be developed and leveraged which allow designers to explore large spaces of design concepts that would be difficult or impractical without computational support. For example, using this scoring an algorithm could search extensive design catalogues for compatible solutions for each of the given functions of a model, a process that would be painstakingly tedious for a team of designers to perform. This paper presents a general framework for integrating the RISCS scoring with both types of design processes.

The authors previously presented the use of RISCS scoring in [46]. This paper further contextualizes the use of RISCS by showing how it may be used for optimization and concept selection, and by presenting an expanded monopropellant system design case study. To demonstrate how it can enable optimization and concept selection, use and adaptation of this scoring function is shown in the optimization of control features and function structure selection for a monopropellant orbiter. The next sections present background in function modelling and fault simulation, discuss research context and definitions of resilience, introduce and construct the scoring function, present how the function may be used for concept selection and optimization, and demonstrate the approach by applying it to design and optimization of a monopropellant orbiter.

2 Background

This paper relies on previous work in functional modelling and the IBFM toolkit to generate a fault model and existing definitions of resilience to construct a scoring function. Both are discussed below.

2.1 Functional modelling

Functional modelling is a way of representing the concept of purpose in a system which has been described as a language for conceptual design intention, a bridge between high-level decision-making and implementation [47], and a "blueprint" for the future system which is agnostic of any particular form [48]. While a variety of modelling conventions have been presented in general, function modelling represents a system as a set of functions which act on flows of energy, material, and signal to accomplish a given task [47]. This specific representation of functionality is one of many system representations of products, but is uniquely useful for its lack of ambiguity and ability to be reused and transformed to simulate behavior [49]. It has long been a part of the engineering design curriculum [50] [45] [51] and has subsequently been standardized as a part of the systems design process [52] [53].

The representation used in this paper follows the convention described in [48]. Flows can represent any sort of material, energy, or signal which passes through the system, while functions represent any operations that happen to the flows on their way through the system which are necessary for accomplishing the overall purpose of the model, which are stated as verb-noun pairs. Using this approach, a typical way to develop a functional model is to create a black-box model of the system which states the overall function with all of the known flows going in and out of the black box. The designer then creates function chains by "following the flow," identifying and sequencing the operations that must be done to the input flow to transform the input flow into the output flow. Finally, the function chains are aggregated and connected as needed to create the overall functional model.

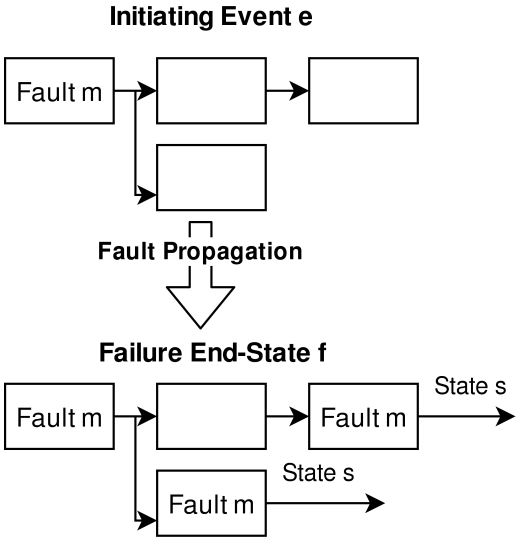

Fig. 1. Illustration of a fault propagation simulation using IBFM. A fault propagates from an initiating mode through the flows of the functional model until it produces an end-state with resulting fault modes and flow health states.

2.1.1 IBFM simulation and model definition

This paper uses the Inherent Behavior in Functional Models (IBFM) modeling language to simulate the fault propagation behavior of a system given its functional model. IBFM was developed for this purpose, with a focus on speed and ability to simulate large sets of joint-fault scenarios [33]. This tool acts by constructing a behavioral model from the functional model by associating behaviors with functions and states with flows. The resulting behavioral model can be used to generate a set of failure scenarios based on individual or combinations of fault events which are then propagated through the model until an end-state is reached. Because this propagation runs as a state machine (and not a set of differential equations), the simulation allows for many sets of faults to be run quickly. Important related terminology is listed below as it integrates directly with the framework presented in this paper:

Fault scenario: A particular instance of the IBFM model for one specific fault event (notated e). IBFM generates a list of fault scenarios E depending on the number of joint faults considered in a run.
End-State: The result of a fault event being run to completion f, inclusive of the final modes and flow states.
Mode: A state of a function m which is associated with a particular behavior. Modes are either fault modes, implying an undesired behavior, degradation, or loss of function, or nominal modes, implying that the function is operating as desired. In this paper, we differentiate two types of fault modes: *initiating fault modes* (in e), which are the initial faults which generate a scenario, and *conditional fault modes* (in f), which happen as a result of other faults.
Condition: A rule that specifies a function's change in mode as a function of incoming flow health state.
Behavior: A property of a mode which determines how incoming flow health states determine outgoing flow health states.
Health State: The quality of the flow represented as a combination of properties $\vec{s}$ given to rate and effort variables that may take the value zero, low, nominal, high, or highest.
Rate Variable: A state of flows representing rate, a concept analogous to throughput or velocity.
Effort Variable: A state of flows representing effort, a concept analogous to force or pressure.

To illustrate what these terms mean in the context of an IBFM simulation, a simulation of a single fault scenario is shown in Figure 1. As can be seen, a scenario starts with a fault event–a set of initiating fault modes which change the flow states output from the functions. After the health states are propagated through the model, the simulation settles at an end-state comprising the final state of the flows and the conditionally-triggered fault modes.

2.2 Resilience

A number of definitions of resilience have been introduced across a variety of fields, including ecology [54] [55] [56], psychology [57] [58], economics [59] [60], sociology [61], network science [62] [63] [64], and management [65] [66] (particularly the management and design of supply chains [67]). Resilience is broadly defined as the ability of a system to prepare for, absorb, recover from, and adapt to failure events, and resilience strategies typically focus on the temporal adaptation to failures, as opposed to risk management, which is more focused on preventing the failure events [68]. However, definitions and metrics for measuring resilience vary across and within fields, with both qualitative and quantitative metrics

[69]. Key dichotomies include:

Engineering resilience and ecological resilience: In engineering resilience, the performance and stability of the system state is recovered to the original system state while in ecological resilience the function of the system is recovered from a static failure state to a new dynamic state, potentially as a result of a change in components (e.g. similar species taking the role of a newly-extinct species) [55].

Deterministic and probabilistic measures: Probabilistic measures consider the uncertainty of disruptions or failure events while deterministic measures do not [69].

Dynamic and static measures: Dynamic measures take into account time-dependent behavior while static measures do not [69].

Of particular interest to this paper is how to use the concept of resilience to motivate design decision-making in engineering design. As with the broader fields of science, definitions and metrics of engineering resilience vary; however, the time-based response to disruptive events is key [70]. Within engineering design, recoverability, defined as a product of the diagnosis capability, resource availability, and repair capability of the system over time has been proposed as an indicator or overall system resilience [71]. While these metrics capture resilience as a metric, they are incomplete as design metrics because they do not incorporate the trade-offs between resilience and other cost and performance considerations–the goal of the metric presented here.

2.3 Decision-based Design

In order to create a metric suitable for design which resolves the trade-offs between resilience and other cost and performance considerations, this paper relies on previous work in decision-based design. Decision-based design is an engineering framework which views engineering as a decision-making process analogous to known work in decision theory [72]. These frameworks rely on the axiomatic definition of utility presented by Von Neumann and Morgenstern in [73]–seeking to maximize the statistical expectation of the utility of a design [74] [72] [75]. This utility is often calculated directly as a profit value (as in [76]), but may often be a function of a profit value, when different profit levels result in different marginal utilities [73].

Decision-based design approaches risk in design as a lottery. To illustrate, if a design x could lead to only two possible outcomes, 1 and 2, with utilities $u_1(x)$ and $u_2(x)$, and outcome 1 has probability $p(x)$, the expected utility $U(x)$ of that design is:

$$U(x) = p(x) * u_1(x) + (1 - p(x)) * u_2(x) \tag{1}$$

Considering this, a design where $p(x) = 0.5$, $u_1 = 200$, and $u_2 = 0$ is equivalent to a design where $p(x) = 1.0$, $u_1 = 100$, and $u_2 = 0$, since both have the same expected utility of $U = 100$. Alternatively, a design where $p(x) = 1.0$, $u_1 = 100$, and $u_2 = 0$ would be preferred to a design where $p(x) = 0.9$, $u_1 = 100$, and $u_2 = 10$, since it has a higher total utility ($100 > 91$).

Because Decision-based design is ultimately a quantitative framework, it lends itself to optimization [76]. For industry applications, this is generally approached by balancing a cost model against a consumer-choice-based demand model to determine how profitable a design will be, and, consequentially, how much utility it provides [77]. Within the systems engineering community, it has developed into the value-driven design framework, which seeks to quantify the cost and value of attributes in order to truly optimize a design, rather than impose requirements [78].

Previous work has used similar expected-value based metrics to approach design risk. Expected cost was first proposed as a replacement to the Risk Priority Number in Failure Modes and Effects Analysis in [79], due to inherent definitional flaws which lead to inconsistent risk priorities in existing practice. In [44], expected-cost-based objectives were formed for the allocation of health management within a functional model. Further work has shown how to use expected value metrics based on the quantified the cost of reacting to failure events to represent the value of resilience [80]. Furthermore, [38] developed a function to represent resilience in a function fault-scenario context. This work seeks to build on these approaches by further showing how to integrate design costs, fault probabilities, and scenario costs with a fault model to enable optimization and concept selection

3 Quantifying RISCS: A Scenario-based Scoring for Function-based Fault Models

To aid decision-making between different functional models based on fault simulation information, this paper introduces the Resilience-Informed Scenario Cost Sum (RISCS), a scoring function which incorporates the trade-offs between a system's design costs, operating costs, and failure behavior. Of particular interest is this function's approach to modeling failure behavior, which is built on IBFM's conception of a fault scenario–a set of faults which yield an end-state. The basic form of this function is a sum of the design costs C_D, operating costs C_O, and fault event costs C_E as shown in the following equation:

$$C = C_D + C_O + C_E \tag{2}$$

This scoring is quite similar in form to that devised in [38] in that it considers trade-offs between design and operation costs, and the response of the system to various failure events. However, the main difference is the applicability to design factors and integration with function-based fault model definitions. While the scoring function devised in [38] merely considered the cost of mitigating factors which reduced the probability of end-states, the function introduced here is generally applicable to all mitigating actions and design changes, and is more closely integrated with the results of IBFM simulations, as will be shown in the following sections.

3.1 Design Cost

Design costs resulting from design changes depend on the considered design problem. This is because, in general, design costs come from a variety of sources, including research and development, required materials, procurement, manufacturing, and integration. For design purposes, this paper considers that the design costs of a given functional model can be represented as individual costs within each function. As is the approach with risk and failure modes [16], these costs can in turn be estimated based on an organization's past costs for those functions. In this case, the resulting equation for design cost C_D is then:

$$C_D = \sum_{n \in N} C_n \tag{3}$$

where C_n is the cost of a given function n in the set of all function instances in the model N.

3.2 Operation Cost

As with design costs, operating costs must be estimated based on an expectation of the system which may result from past performance or a designer-created parametric model specific to the problem. While the method for determining the operating costs will vary depending on the considered problem, in general they relate to the individual flows going in and out of the black-box form of the model. Flows going into the model can result in costs (such as those having to do with raw materials and energy) and revenues (such as those that take in waste material), as do flows going out of the model, with costs potentially resulting from waste streams and revenues resulting from useful goods created. In general, the operational cost C_O then follows the form:

$$C_O = \sum_{l \in L} C_l - R_l \tag{4}$$

where C_l and R_l are the respective costs and revenues associated with the flow l in the set of inflows L entering and leaving the black box model.

It should be noted that, as will be the case in the case studies, these C_l and R_l terms need not stand for *explicit* costs and revenue in dollars generated, but can also stand for the normative goods, utility, or externalities created and destroyed by the organization. That is, if the organization has a normative goal (e.g., generate science, provide public infrastructure, etc) that does not explicitly lead to more or less revenue, the goods created or destroyed by the organization by pursuing this goal can be quantified and incorprated as if it were a direct cost or revenue.

3.3 Fault Scenario Cost

Key to representing resilience in a system is the time-based response to a large number of disruptions or threat vectors [81]. This paper's incorporation of these disruptions is based around IBFM's ability to simulate large numbers of fault events, which are created by initiating a set of fault modes which propagate through the model until an end-state is reached. These events have costs, which are determined from the system's response to the events in terms of end-state flow-states and modes, and probabilities, which are determined by the probabilities of the initiating fault modes. The cost of fault events is calculated as an expected cost–the cost of each event is weighted by its probability. The resulting fault event cost C_E follows the general form:

$$C_E = \sum_{e \in E} P_e * C_e \tag{5}$$

where P_e and C_e are the respective probabilities and costs of fault event e in the set of considered events E. Assuming the probability of individual fault modes in a scenario are independent, the probability of a given scenario is simply the product

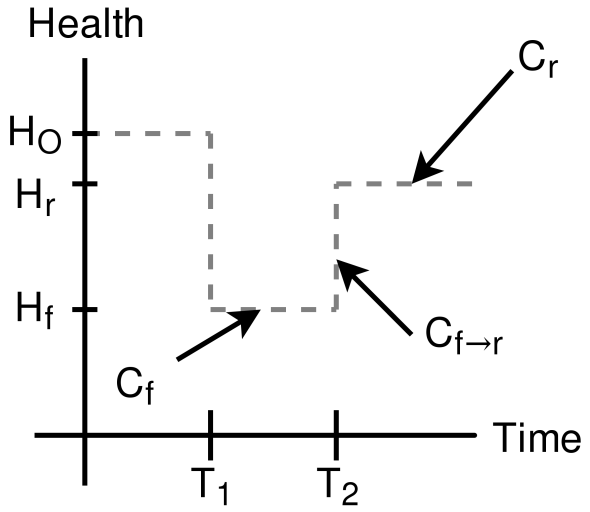

Fig. 2. Costs associated with a failure event in a resilient system.

of the probability of the specific combination of originating faults occurring multiplied by the probability that the rest of the system remains nominal as follows:

$$P_e = \prod_{m \in e} P_m * \prod_{n \notin e} (1 - \sum_{m \in n} P_m) \tag{6}$$

where P_e is the probability of a given fault event e, P_m is the probability of a given initiating fault mode m in event e occurring, and n is a function that does not have a fault in the event e.

As an expected cost metric, this fault scenario cost bears structural similarities to the Risk Priority Number (RPN) used in Failure Modes and Effects Analysis (FMEA), in that it multiplies the probability of a fault with the severity of the consequences of the fault. Aside from that basic similarity, there are a number of differences between an expected cost metric and an RPN, as listed in [79]. Unlike the RPN, expected cost uses real probabilities and costs, making it a consistent metric for risk prioritization, and a valid metric to trade off with with design and operational costs. While the criticality rating used in Failure Modes, Effects, and Criticality Analysis is in general a consistent risk prioritization metric [79], it still lacks the key property of allowing trade-offs with design and operational costs.

To further consider resilience in an expected cost setting, this work extends the definitions of expected cost by considering three distinct stages in the system's failure and recovery which in turn map to three distinct costs. These costs are, as shown in Figure 2: the cost of failure state C_f, the cost of mitigating or repairing the failure $C_{f\rightarrow r}$, and the cost of partial recovery C_r. This definition lines up with common definitions of resilience, in which the system starts at a nominal stable condition, enters an unstable state due to a disruption, and then settles in a new recovered stable condition [82]. The resulting fault event cost follows the form:

$$C_e = C_f + C_{f\rightarrow r} + C_r \tag{7}$$

These cost definitions integrate with IBFM simulations as follows:

C_f results from the scenario end-state of the associated initiating fault event e,
C_r results from the scenario end-state of a new fault event simulation, with a chosen set of modes repaired, and
$C_{f\rightarrow r}$ results from the cost of repairing the modes present in the end-state of the associated fault event e not used as fault modes in the recovered fault event simulation.

This process of running a failure scenario, selecting modes to repair, and running a new fault simulation based on the unrepaired modes is shown in Figure 3. It should be noted that this fault re-simulation (and resulting cost) is only necessary for the special case in which only a partial recovery is attempted or possible. When all of the modes are recovered (i.e. a full recovery), no new fault simulation is needed since the end-state will be nominal, resulting in zero partial recovery cost C_r. Alternatively, if it is impossible to repair the modes and/or no recovery is attempted, no additional cost results from recovery or partial repair–instead the costs merely result from the failure state. Calculating each of these costs is discussed in the following sections.

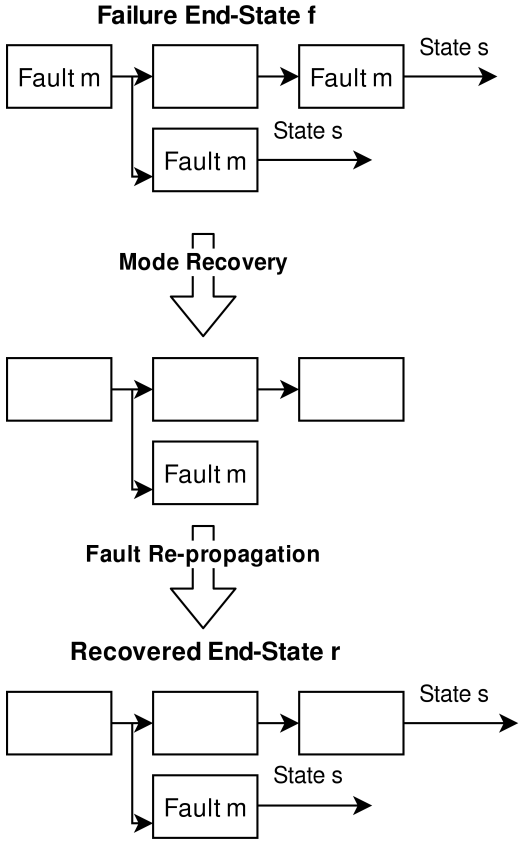

Fig. 3. Illustration of fault re-simulation required to capture the costs of partial recovery C_r.

$$C = \sum_{n \in N} C_n + \sum_{l \in L} C_l - R_l + \sum_{e \in E} \left(\prod_{m \in e} P_m * \prod_{l \notin e} (1 - \sum_{m \in e} P_m) * \left(\max_{m \in f \cap \notin r} (t_m) * \sum_{l \in L} \bar{c}_l[\vec{s}_{f,l}] + \sum_{m \in f \cap \notin r} C_m + t_r * \sum_{l \in L} \bar{c}_l[\vec{s}_{r,l}] \right) \right) \quad (8)$$

Table 1. Cost rate of an individual flow state for flow l based on combination of rate and effort health states.

Effort	Rate Health				
Health	Zero	Low	Nominal	High	Highest
Zero	$\bar{c}_l[00]$	$\bar{c}_l[01]$	$\bar{c}_l[02]$	$\bar{c}_l[03]$	$\bar{c}_l[04]$
Low	$\bar{c}_l[10]$	$\bar{c}_l[11]$	$\bar{c}_l[12]$	$\bar{c}_l[13]$	$\bar{c}_l[14]$
Nominal	$\bar{c}_l[20]$	$\bar{c}_l[21]$	$\bar{c}_l[22]$	$\bar{c}_l[23]$	$\bar{c}_l[24]$
High	$\bar{c}_l[30]$	$\bar{c}_l[31]$	$\bar{c}_l[32]$	$\bar{c}_l[33]$	$\bar{c}_l[34]$
Highest	$\bar{c}_l[40]$	$\bar{c}_l[41]$	$\bar{c}_l[42]$	$\bar{c}_l[43]$	$\bar{c}_l[44]$

3.3.1 Failure Cost

Failures result in costs because they degrade the important flows leading in and out of the system, resulting in higher costs, less revenue, or less utility. To determine the costs of specific failure events, specific costs must be associated with the flow health states present in the end-state of the fault event. These flow states are defined as the quality (zero, low, nominal, high, or highest) of a flow's rate and effort components. The general form of a specific matrix $\bar{c}_l$ for flow l is shown in Table 1. Note that these, as *specific* costs, are the cost per unit time until the failure is mitigated, as the total cost of a failure depends both on both the severity of the failure state and the amount of time the system is in the failure state. As was discussed in Section 3.2, these important flows must be identified by the designer with costs included.

The cost of the failure part of the fault event can then be calculated using the state of the flows going in and out of the model and the time taken to mitigate the failure as follows:

$$C_f = t_f * \sum_{l \in L} \bar{c}_l[\vec{s}_{f,l}] \tag{9}$$

where C_f is the cost of the failure scenario end-state f that is the direct result of the simulation of fault event e, t_f is the time taken between the failure and the recovery, l is a given flow in the set of input and output flows L, $\bar{c}_l$ is the specific cost matrix for that flow, and $\vec{s}_{f,l}$ is the end-state of that flow l in the given scenario f.

The recovery time is the time necessary to repair the individual failure modes which are in the failure scenario but not in the recovered scenario. Considering that the repairs may be done in parallel, this time can be calculated as the maximum of the times t_m needed to repair each failure mode m in the failure scenario end-state f but not in the recovered scenario r.

$$t_f = \max_{m \in f \cap \notin r} (t_m) \tag{10}$$

3.3.2 Mitigation Cost

The cost of mitigation is a result of repairing the failure modes in the failed system scenario that are not present in the recovered system. This cost $C_{f \to r}$ is calculated as the sum of the cost C_m of recovering each mode m which is present in the failure scenario end-state f but not in the recovered scenario r, per the following equation:

$$C_{f \to r} = \sum_{m \in f \cap \notin r} C_m \tag{11}$$

As with the other costs, this mode recovery cost C_m can be estimated based on past data or assumptions about the future system regarding the repair/replacement processes required for each function.

3.3.3 Partial Recovery Cost

Finally, the cost associated with the recovered system is the cost of the degraded flows still present in the end-state recovered system due to unrepaired or unrecoverable failure modes. This may be calculated similarly to the failure cost, by running a new fault scenario using the failure modes in the recovered state, as shown in Figure 3 This partial recovery cost C_r is a result of the time left in the recovered state (i.e., for the remaining life of the system) t_r and the specific costs $\bar{c}_l$ of the state $\vec{s}_{r,l}$ in the recovered end-state r of the flow l in the set of input and output flows L. This is shown in the equation:

$$C_{r,R} = t_r * \sum_{l \in L} \vec{c}_l[\bar{s}_{r,l}] \tag{12}$$

These costs are calculated over the rest of the life of the system. If the system is meant to operate for a long time, a discount factor should be applied based on the time value of money for the organization.

3.4 Summary

The previous sub-sections discussed how to calculate the costs of a system considering the design, operational, and fault scenario costs using the IBFM simulator, with special consideration of resilience–the ability to recover from failures. While this metric may be calculated differently depending on the considered design problem, when stated as a single expression using the constructions developed in the previous sections, the equation for RISCS takes the form shown in Equation 8, where:

C is the total RISCS
C_n is the cost associated with the design of a function
n is a function
N is the set of function instances in the model
C_l is the cost associated with an input or output flow
R_l is the revenue or utility associated with an input or output flow
l is a flow
L is the set of input or output flows
e is a fault event, a combination of fault modes
E is the set of considered fault events

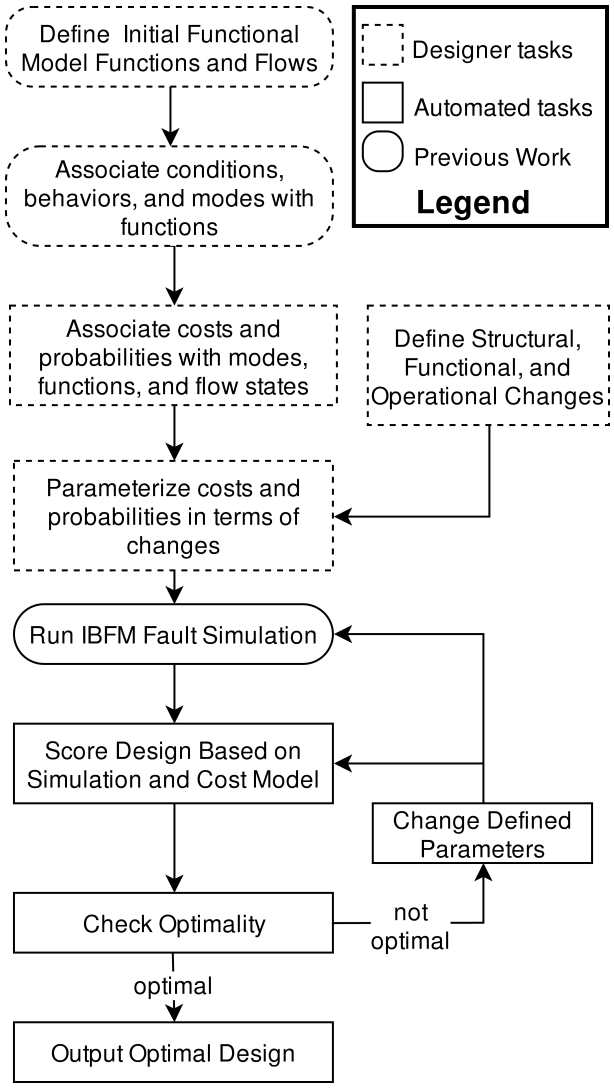

Fig. 4. Framework enabled by integrating cost-based scoring and fault simulation. The designer sets up a parameterized design problem which is then solved by an optimization algorithm.

m is a fault mode
P_m is the probability of a fault mode
f is the resulting fault scenario end-state of the fault event e
t_m is the time taken to repair a mode
r is the recovered scenario that is the result of repairing fault modes in f
$\bar{c}_l$ is the cost function of the flow based on its state
$\vec{s}_{f,l}$ is the state of the flow l in the scenario end-state f
C_m is the cost associated with repairing a fault mode
t_r is the time remaining between recovery and the end of life of the system
$\vec{s}_{r,l}$ is the state of the flow in the recovered scenario

4 Applying RISCS to Optimization and Concept Selection

This paper proposes and demonstrates the use of the RISCS score enumerated in Section 3 for functional model concept generation and selection. As will be discussed in this section, this can be performed in two ways: in an optimization procedure in which the RISCS score is parameterized over a defined space of variables of interest or to evaluate and compare different concepts of interest to the designer. While following subsections discuss both of these applications and and demonstrate them in the design and selection of resilient features in a monopropellant system, an elementary example is provided first to provide conceptual insight.

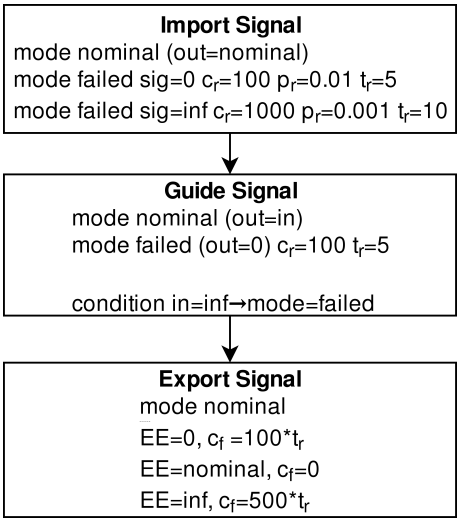

Fig. 5. Functional model of a signal-carrying medium, with modes, conditions, costs, and probabilities associated to each function.

4.0.1 Elementary Example

To briefly illustrate the use of RISCS scoring, the design of a signal-carrying medium is considered, as shown in Figure 5 with modes and conditions as they would be modeled in IBFM (albeit for forward propagation of faults only, without separate rate and effort states). As shown, there are two initiating failure modes in the `Import Signal` function (no input signal and an infinite (shock) signal) and one condition and resulting conditional mode in the `Guide Signal` function (a condition specifying that the function enters the failed mode when a shock signal is received, and a failed mode which makes the output signal zero if it receives a shock signal). The resulting failure costs are shown in the `Export Signal` function, based on the system exporting a zero, nominal, or shock signal.

To illustrate how RISCS scoring can aid design decision-making, two designs will be considered: one in which the `Guide Signal` condition exists (Design 1), and one in which it does not (Design 2). These designs each have two scenarios resulting from the initiating fault modes: the scenario in which the incoming signal is failed with a value of zero, and one in which a shock enters the signal (e.g. from a short circuit). In the first scenario, the expected cost is the same for both designs. The zero signal state propagates from the `Import Signal` function to the `Export Signal` function, resulting in a cost of failure resulting from the cost rate of the failed flow state, the time needed to repair the fault, and the cost of repairing the fault:

$$\begin{aligned} C_{S1} &= P_{f1} * (C_f * t_r + C_r) \\ &= 0.01 * (-100 * 5 - 100) = -6 \end{aligned}$$

In the second scenario, the expected cost is different for each design, due to the differences in fault propagation. In the first design, the shock causes a failure mode in the `Guide Signal` function, resulting in a zero flow state in the `Export Signal` function, and costs from the failed (zero) flow state and the times and costs needed to repair the `Import Signal` and `Guide Signal` functions. In the second design, the shock propagates through the `Guide Signal` function, resulting in costs from a failed (inf) flow state and the time needed to repair the function. The resulting expected costs are tabulated for each design below:

$$\begin{aligned} C_{S2} &= P_{f2} * (C_f * t_r + C_r) \\ &= 0.001 * (1000 + 100 + 100 * 10) = -2.1 \\ &= 0.001 * (1000 + 500 * 10) = -6 \end{aligned}$$

When the expected failure costs are combined with design and operational costs, the overall costs can be tabulated. In this example problem, it is assumed that the designs had the same operational costs, and design costs of -2 and -1, respectively. The resulting cost score is tabulated in Table 2. As can be seen, while Design 1 has slightly more design cost, the lower expected cost resulting from *not* propagating the shock flow results in a better overall cost score. Given different numbers for probabilities, repair times, and costs, however, this result would change. For example, if the `Guide Signal` function had a repair time of 100, the resulting repair time would make the expected cost of failure of Scenario 1 for Design

Table 2. Cost comparison of Design 1, as shown in Figure 5, and Design 2, with the condition removed.

Costs	Design 1	Design 2
Design	-2	-1
Operational	20	20
Scenario 1	-6	-6
Scenario 2	-2.1	-6
Total	9.9	7

$1 - 11.1$, making the overall cost score worst than that of Design 2. Alternatively, if said design had commensurately lower design or operational costs, this design would still be superior using this method. This demonstrates how RISCS scoring can be used to trade off the modeled dynamic failure response, failure costs and probabilities, and design and operational costs to determine if a resilient feature should be added to a design.

4.1 Optimization

To optimize functional models based on the RISCS score, this paper proposes the general framework shown in Figure 4: the designer defines an initial functional model and creates a behavioral model by associating conditions, behaviors, and modes with the various functions as presented in [33] using IBFM. The designer then associates costs and probabilities with the various modes, functions, and flow states, and defines the changes to be explored within a parameterized cost scoring. This cost scoring is then optimized by an appropriate algorithm by running the fault simulation, scoring the design, and changing the parameters until an optimal design is found. This enables the designer to explore a large space of design alternatives in a systematic, automated way without a tedious investigation of every model variant. This approach is demonstrated in Section 4.3.1 and discussed below.

In order to explore a large space of design changes, the RISCS must be parameterized over a space of possible design changes that may be made to the model. A variety of design changes which may be pursued in the context of function-based fault modelling have been presented previously in [38] [40], which are compiled along with changes pursued in other function-failure optimization approaches ([44] [43]) along with new design changes which have been identified by the authors in Table 3. As can be seen in the right side of the table, each design change has associated potential difficulties which may make it difficult to effectively model or predict its effect, but also may provide value depending on the design problem considered. A key difficulty common to many different design changes is predicting how a change will effect the design and operating costs of the functional model given the different couplings which may occur, for example, due to the working, constructional, and system interrelationships which are developed later in the design process. Such couplings are prevalent in the embodiment design stage of highly-coupled engineered systems, and provide significant challenges to design coordination [83].

However, due to the principal nature of the functional representation of the design, especially in the context of risk, many of these concerns may be considered to be second-order in nature unless it is known that the future design will be a highly coupled system. In general, the parameters to be optimized in these systems are either structural (which sizing and configuration do not influence, but instead flow out of), high-level features to be added (which do not necessarily impact performance), or operational (with very little coupling). Depending on the design case, the designer may choose to use the informal approach to only parameterize a subset of the design problem that can be readily modeled. Alternatively, sufficient resources (e.g. design catalogues, known physical effects, or cost models) may be available for certain problems, depending on how much is already known about the design space. While many of the challenges presented in Table 3 can be avoided depending on the context and purpose of the optimization, future work should provide a means to address them in the general case.

Additionally, optimizing each of the design changes shown in Table 3 will require different solution strategies depending on the variable type. Structural design variables such as redundancy, function order, and flow paths will likely need to be approached using a graph grammar-based computational synthesis framework, as has been shown in [84] [85], since the function structure is fundamentally a graph. However, the internal-functional or operational parameters may require different algorithms depending on how RISCS is represented and parameterized. For example, a gradient-based search may be performed over the parameters of the assumed realization of the function (e.g. sizing, quality, etc), while an evolutionary or direct search method is used to determine the modes to recover or conditional logic. Because of the different algorithms that lend themselves to each design change, optimizing multiple variable types at once may require a specialized solution strategy such as a multidisciplinary design optimization [86] to allow each parameter to be optimized appropriately.

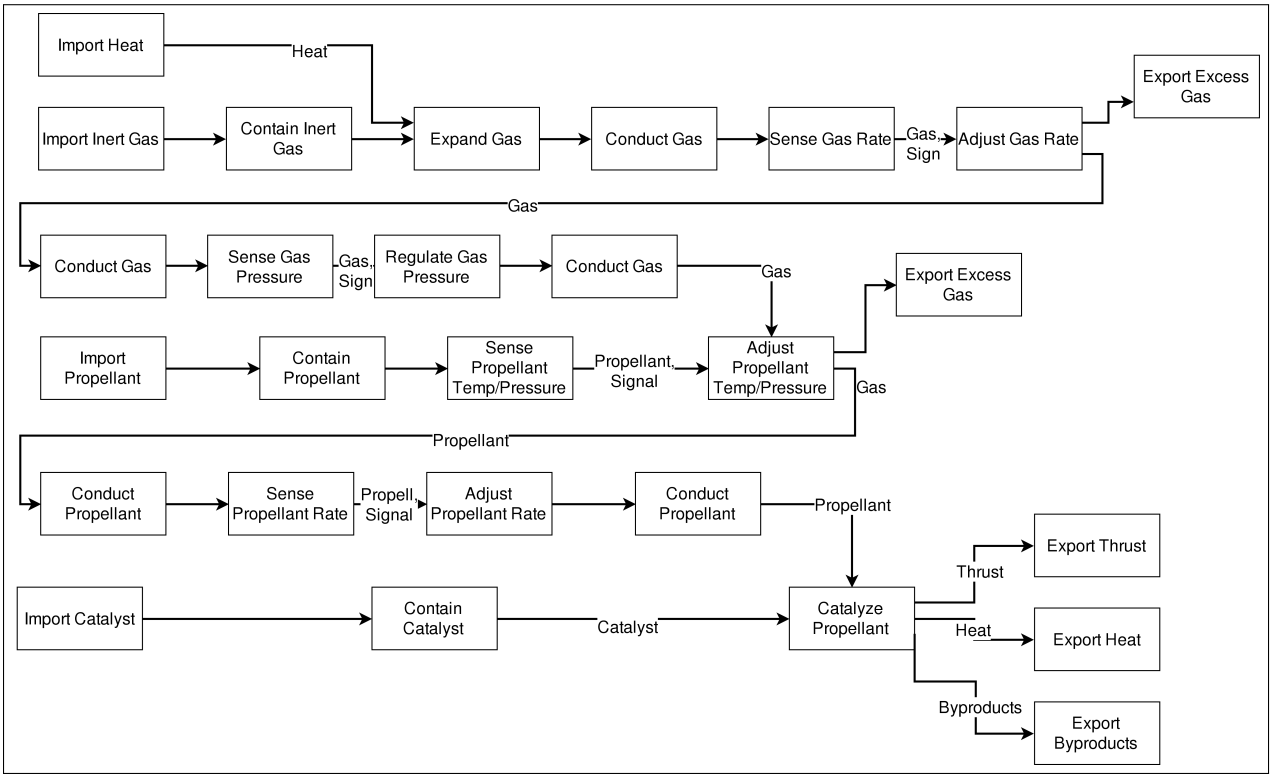

Fig. 6. Functional Model of Base Monopropellant System

4.2 Concept Selection

When the space of designs is difficult to parameterize, the RISCS score may instead be used to compare and select individual solution concepts. For example, totally different solution concepts may be difficult to credibly generate and evaluate in an optimization process if the costs and modeled behavior of functions changes depending on the intended purpose in the model. When the designer encounters this problem, or simply wishes to give more attention to how individual solution concepts would work, they may instead use the following process to generate and select designs:

1. Define the functional concepts to consider
2. Construct fault model by associating conditions, behaviors, and modes with each concept's functions
3. Associate costs and probabilities with modes, functions, and flow states for each concept
4. Run IBFM fault simulations for each and tabulate RISCS score
5. Choose the best-scoring concept

This process is essentially the same as the the optimization process shown in Section 4, except instead of parameterizing the space of changes and applying an algorithm, each of the concepts are generated individually at the beginning. However, in this process there is more ability for the designer to gain insight into the design problem itself, gain knowledge to influence future designs, and devise new features or re-think the design problem, as they receive direct feedback from the model. This approach is demonstrated in Section 4.3.2 to compare resilient features in the design of a monopropellant system.

4.3 Case Study: Design of Monopropellant System

The following section demonstrates application of RISCS to the design of a monopropellant orbiter, a system previously considered in [38] and introduced as an example system in [87]. Monopropellant propulsion systems are named as such because they do not require a separate oxidizer, and are commonly used in spacecraft for attitude control, and sometimes to provide primary thrust. The functional model of the monopropellant system is shown in Figure 6. Heat is applied to an inert gas to expand, and the gas is regulated to an appropriate temperature and pressure. The expanded gas then pushes a propellant over the catalyst. As propellant passes over the catalyst, it reacts, resulting in thrust.

The overall value generated by this system is a result of the quality of the thrust function, and the costs are a result of any design costs incurred by each function. In the formulation of the RISCS score considered in this application, only these trade-offs are considered for simplicity to illustrate the approach. As a result, only a few components must be considered in the cost function (design and failure state costs), since operational costs are assumed to be constant between concepts. Additionally, because of the context of the system, which operates in space, where there is no ability to repair or maintain

the system, the repair costs and future state costs are not included in this function. The resulting cost function is:

$$\sum_{n \in N(\vec{x})} C_n(\vec{x}) + \sum_{e \in E} P_e * (\bar{C}_l[\vec{s}_{f(\vec{x}),l}]) \tag{13}$$

where the notation is consistent with that outlined in Section 3, except P_e is the probability of an event (which does not change with changes in condition), t_m is an (assumed constant) mitigation time, E is the set of single fault (and no-fault scenarios), $\bar{C}_l$ is the cost matrix for the thrust function (the only desired output flow) which has values shown in Table 4 which are again scaled to consider different failure costs, and $\vec{x}$ is the given design.

4.3.1 Optimization of Controlling Functions

This section applies the RISCS function to the optimization of controlling functions within the functional model of the variant of the monopropellant system shown in Figure 14 in Appendix 6. Controlling functions refer to the functions in the model which change the response of the system based on a signal indicating a change in flow. In this study they represent the high-level requirements for the control systems of the regulating functions in case of a degradation or failure in the upstream flows. That is, they represent whether the system should be designed to recover a flow (which would compensate for the failure but increase initial design costs) or keep the flow state constant. In the model of the monopropellant propulsion system, these functions are *control gas rate*, *control gas pressure*, *control propellant temp/pressure*, and *control propellant rate*. When the system is realized, these might be manifested as logic gates, control circuitry, or any system which takes actions based on an input. This is represented in IBFM as changes in conditions which cause the system to enter modes with different behavior.

```
mode 1 Operational EqualControl
mode 2 Operational IncreaseControl
mode 3 Operational DecreaseControl
condition 1 3 to 2 LowSignal
condition 1 2 to 3 HighSignal
condition 2 3 to 1 NominalSignal
```

Fig. 7. Example controlling function conditions and modes.

To illustrate, in the function definition shown in Figure 7, the modes `EqualControl`, `IncreaseControl`, and `DecreaseControl` each refer to behaviors in which the controller keeps the incoming flow state, increases the incoming flow state, and decreases the incoming flow state, respectively. Similarly, the conditions `LowSignal`, `HighSignal`, `NominalSignal` refer to a lower-than-nominal, higher-than-nominal, and nominal flow state, respectively. Finally, the conditional logic specifies which mode to enter based on the condition. For example, `1 3 to 2` means the function increases the flowstate by entering mode 2 (`IncreaseControl`) when it was previously in mode 1 (`EqualControl`) or 3 (`DecreaseControl`).

This problem is readily encoded as an integer vector and can be solved using an evolutionary algorithm following the general optimization framework shown in Figure 4. Although many integer programming approaches are possible (indeed, a direct search method may be more efficient, and the space of designs is small enough to be searched with a brute force method), the evolutionary algorithm was used in this paper in which the initial population is initially seeded with the solution of `EqualControl` for each state of each condition in each function to speed the solution process.

As can be seen in Figure 8, the use of this algorithm increases the overall RISCS significantly from the baseline design cost. This results in a design shown in Figure 5. As can be seen, while expensive recovery options are avoided (such as attempting to increase the final flow of propellant in Controller 4), less expensive recovery features are added in order to achieve the best RISCS score. This demonstrates the ability of the RISCS score to integrate with an optimization framework to systematically explore the space of design variants.

4.3.2 Comparing Model Structures

This section applies the RISCS function to compare different design concepts for adding resilience to the monopropellant system. Each of these variants are shown in Appendix 6, and were constructed by adding additional functions, conditions, and behaviors to the baseline IBFM model to account for the added resilient feature. The design variants considered were:

1. **Redundant Gas Tanks:** The *Contain Inert Gas* and *Expand Gas* functions are made redundant, as shown in Figure 10.

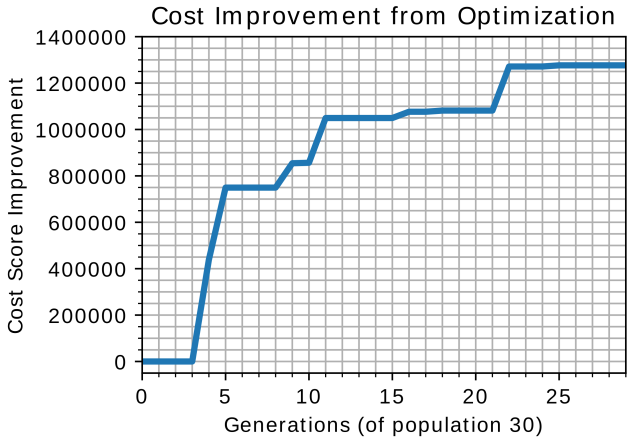

Fig. 8. Cost optimization of the functional model using the evolutionary algorithm, showing how value can be increased using the presented optimization framework.

2. **Redundant Thrusters:** The *Contain Catalyst* and *Catalyze Propellant* functions are made redundant, as shown in Figure 11.
3. **Auxiliary Heat-recovery system:** An auxiliary system powered by the heat from the combustion from the propellant is added to expand the gas in case the heat source is lost, as shown in Figure 12.
4. **Redundant Pressure Regulators:** An additional pressure regulator is added which activates when the operating pressure regulator fails, as shown in Figure 12.
5. **Optimized Control Features:** The control features optimized in Section 4.3.1 are used in the design, as shown in Figure 13.

As can be seen in the functional models for each design variant, each resilient features adds initial design cost. In order to calculate the design scoring in Equation 13, however, each model must be run to determine the resulting failure costs. The results for failure costs resulting from each of these model runs is shown in Figure 9, along with the initial design costs and total costs. For clarity, these are displayed as differential costs from the baseline, to show which features "pay off" by reducing failure costs above their increase in design cost, and which do not.

As shown in Figure 9, as simulated in the model, Variant 1 and Variant 2 (redundant gas tanks and redundant thrusters, respectively), are not worth their design cost because they negligibly improve the overall failure cost. This is because, in the model, these systems have a low probability of faults and relatively high design cost. On the other hand, the features in Variant 3, 4, and 5 do pay for themselves in terms of failure cost. In Variant 3, because the *Import Heat* function was modeled with a relatively high fault probability, the heat recovery feature was able to reduce the impact of this fault substantially. In Variant 4, the redundant pressure sensor is able to increase the scoring function simply because the design feature (a sensor) is relatively inexpensive, even though the change in failure cost is low. Finally, in Variant 5, while the design feature is relatively expensive, it is able to reduce a large amount of failure cost, allowing the feature to justify itself in terms of the cost score.

4.4 Discussion

The previous sections presented and demonstrated approaches for the RISCS scoring function for optimization and concept selection. In both of these approaches, using the RISCS score enables design decision-making based on a fault model by incorporating cost and risk information. For optimization, RISCS scoring is helpful in that it provides a comprehensive resilience-informed objective function that can be optimized. As discussed, this optimization process may only occur, however, when the RISCS score may be parameterized over a space of design changes. While these variables are provided, along with a simple example of using the optimization framework on a narrow space of changes, creating a broad parameterization of a model that is accurate may be difficult. Indeed, even in this case study, assumptions were made about the realized future system which may or may not turn out to be correct. For example, while it was assumed in the case study that each flow could be recovered by one health state by each controller, there may be health states that are unrecoverable, leading to poor modelling of fault propagation behavior and an overly optimistic decrease in failure costs. Alternatively, the control features may simply be impossible or more expensive than expected to realize, leading to an under-estimation of costs. These issues will likely provide difficulties to applying the framework depending on the problem considered. Nevertheless, when the designer does have accurate information, the framework provided here allows them to find the optimal version of their concept, allowing risk reduction to happen earlier in the design phase.

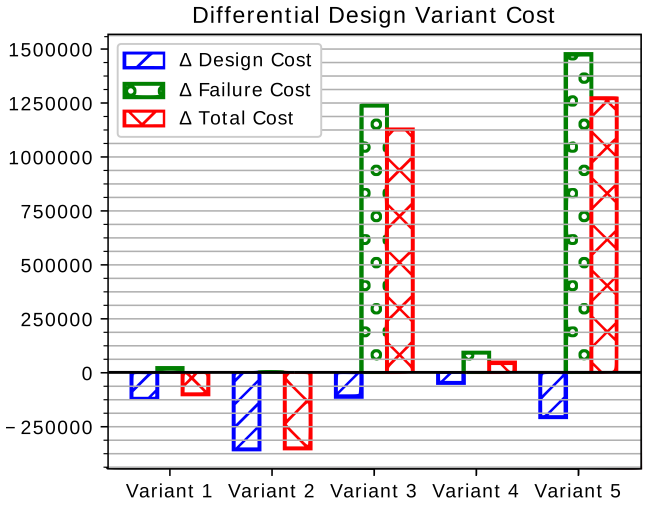

Fig. 9. Differential costs of design variants based on fault simulation.

For concept selection, RISCS scoring gives the designer a comprehensive decision-theoretic metric to compare between two concepts encoded in function-based fault models. In Section 4.3.2, this metric was used to compare design variants that had different resilient features, such as redundancies, recovery systems, and the optimized control system. While this scoring was able to show which features were worth the cost, deciding which set of features to include requires further analysis. This is because it is unknown how these features would interact with each other. For example, while the heat recovery system and controlling features both increase the value of the design, using both features in the same system may be inappropriate if both of them are reducing the same sources of risk, since each would reduce less failure cost if the other system was assumed to be there. As a result, when using RISCS to compare design variants, the designer may need to generate new variants with joint features with a high RISCS score to find the best configuration of features. Finally, while this RISCS scoring was used for different variants of the same functional model, it is expected that this design metric could provide more impact when used to compare wholly different design concepts, rather than variations of a single concept.

5 Conclusions

This paper presents a framework for considering resilience in early design using a resilience-informed scenarios cost sum (RISCS) scoring function to resolve trade-offs between the expected design, operational, and fault response costs. RISCS scoring integrates with a comprehensive scenario-based fault simulation which determines the propagation of faults by mapping the costs of the failure event, its recovery, and the recovered states to flow states and failure modes in each scenario. Approaches to using the scoring are presented, both for optimization and for comparing design concepts, which are then demonstrated in the design of a Monopropellant system, first by optimizing resilient control features, and then by comparing design variants with different function structures. Use of this cost score for these applications is shown to help the designer find resilient features which "pay off"–reducing the expected cost of failure enough to justify their design and operational costs.

5.1 Limitations and Future Work

The RISCS scoring presented here has a number of limitations which should be addressed to realize the full potential of design for resilience. While Section 3 constructs RISCS in a way that integrates well with existing function-based fault model definitions, it may need to be constructed differently based on the design problem to most effectively capture design costs and risks. The following are limitations of the presented approach which may be addressed in future work:

Fault Probability Assumptions: As constructed in Section 3, failures are the result of fault events or joint-fault events, the probability of which is determined based on the assumption that each fault has an independent probability. However, in reality, many joint-fault scenarios may occur for which single-fault probability of failure is incorrect. For example, if a meteoroid were to hit the monopropellant system, it might cause a number of joint faults with a single probability for the event that does not derive independently from the functions themselves. While many of these events may be considered as the modeled effects of fault propagation model, incorporation of non-independent joint faults will need to be incorporated in future work to better consider risks. Furthermore, while the assumption that the expected cost is the probability of the fault multiplied by cost holds true for small probabilities, it should be noted that over long

timescales faults with small probabilities may be expected to occur multiple times. In these cases, for repairable faults, the expected number of events occurring over the life-cycle of the system should be used for the expected cost, rather than the probability.

Incorporation of Risk Attitude: Decision-based and Value-driven design frameworks often calculate the cost of an outcome and then calculate the utility of said outcome, to incorporate the designer's attitude towards risk or the marginal utility of cost. The expected cost can then be calculated as the cost which corresponds to the expected utility of a design. While this approach is common, it was not used here because risk neutrality was assumed to be normative and to allow the designer to use cost as a more holistic measure inclusive of other forms of utility not taken into account in the direct cost of risk. However, future work may demonstrate the traditional certainty-equivalent approach in this context to better integrate with typical cost modeling practice.

Cost and Probability Modelling: No approach is included here to determine how to model costs and probabilities in a systematic way to be used with a functional representation. While the authors propose that these may be associated with past realizations of said functions (as is done for risk and failure modes [16]), future work will need to enumerate how this would be performed. Additionally, there may be integration costs that are not a part of individual function costs that may lead to different forms of the design cost function constructed in Section 3, and, as mentioned in Section 3 the time-value of money may effect the calculation of costs of systems which operate over long timescales.

Feasibility and Non-functional Interactions: As discussed in Section 4, the RISCS score can only be used to compare designs based on cost and risk information; it cannot tell the designer which designs are technically feasible. It further is not based on a fully-realized system, but a functional representation, and as a result may not incorporate all interactions, costs or risks that will eventually occur in that realized system. The authors suggest that to account for these issues, the use of this design scoring should happen in isolation, but in conjunction with feasibility studies for the compared design concepts and features, and should further be used such that it can be updated throughout the design process, as more interactions are included in the design, using dynamic design frameworks such as in [88] to approach the resolution of epistemic and model uncertainties.

6 Acknowledgements

This research is in part supported by the NASA Ames Research Center (award number NASA NS295A) as an I/UCRC Center for e-design project (award NSF IIP-1362167). Any opinions or findings of this work are the responsibility of the authors and do not necessarily reflect the views of the sponsors or collaborators.

References

[1] Forum, T. C., 2005. Chernobyl's legacy: Health, environmental and socio-economic impacts. Tech. rep., International Atomic Energy Agency.

[2] Rogers, E., 1986. Report to the President by the Presidential Commission on the Space Shuttle Challenger Accident. Tech. rep., National Aeronautics and Space Administration, Washington, DC.

[3] Congress, U., 2010. The role of bp in the deepwater horizon explosion and oil spill. Tech. rep., House of Representatives Subcommittee on Oversight and Investigations, Committee on Energy and Commerce.

[4] Seife, C., 2003. "Columbia disaster underscores the risky nature of risk analysis". *Science,* **299**(5609), pp. 1001–1002.

[5] US Military Standard, 1980. Procedures for performing a failure mode, effect, and criticality analysis. Technical Standard MIL-STD-1629A, Department of Defense.

[6] Vesely, W. E., Goldberg, F. F., Roberts, N. H., and Haasl, D., 1981. Fault Tree Handbook. Handbook NUREG-0492, U.S. Nuclear Regulatory Commission, Jan.

[7] de Kleer, J., and Kurien, J., 2003. "Fundamentals of model-based diagnosis". *IFAC Proceedings Volumes,* **36**(5), June, pp. 25–36.

[8] Kurtoglu, T., and Tumer, I. Y., 2008. "A graph-based fault identification and propagation framework for functional design of complex systems". *Journal of Mechanical Design,* **130**(5), p. 051401.

[9] Lawrence, E., 2011. System safety analysis and assessment for part 23 airplanes. Tech. Rep. AC 25.1309-1A, United States Federal Aviation Administration.

[10] Wilkinson, P., and Kelly, T., 1998. "Functional hazard analysis for highly integrated aerospace systems".

[11] Committee, S. I. S.-., et al., 1996. "Arp4761 guidelines and methods for conducting the safety assessment process on civil airborne system and equipment". *Warrendale, Pennsylvania: Society of Automotive Engineers*.

[12] Ericson, C. A., et al., 2015. *Hazard analysis techniques for system safety*. John Wiley & Sons.

[13] Delange, J., Feiler, P., Gluch, D. P., and Hudak, J., 2014. Aadl fault modeling and analysis within an arp4761 safety assessment. Tech. rep., Carnegie Mellon University Software Engineering Inst.

[14] Dowries, C., and Chung, P. W. H., 2011. "Hazards in advising autonomy: Incorporating hazard modelling with system dynamics into the aerospace safety assessment process for uas".

[15] Joshi, A., Heimdahl, M. P., Miller, S. P., and Whalen, M. W., 2006. "Model-based safety analysis".
[16] Stone, R. B., Tumer, I. Y., and Van Wie, M., 2004. "The Function-Failure Design Method". *Journal of Mechanical Design,* **127**(3), July, pp. 397–407.
[17] Lough, K. G., Stone, R. B., and Tumer, I., 2006. "The risk in early design (RED) method: Likelihood and consequence formulations". In ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.
[18] Lough, K. G., Stone, R., and Tumer, I. Y., 2009. "The risk in early design method". *Journal of Engineering Design,* **20**(2), pp. 155–173.
[19] Hutcheson, R. S., and Grantham, K., 2012. "Does access to expert knowledge allow students to better assess risk?". In ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. 145–149.
[20] Hollnagel, E., 2017. *FRAM: the functional resonance analysis method: modelling complex socio-technical systems*. CRC Press.
[21] De Carvalho, P. V. R., 2011. "The use of functional resonance analysis method (fram) in a mid-air collision to understand some characteristics of the air traffic management system resilience". *Reliability Engineering & System Safety,* **96**(11), pp. 1482–1498.
[22] Rasmussen, B., and Whetton, C., 1997. "Hazard identification based on plant functional modelling". *Reliability Engineering & System Safety,* **55**(2), pp. 77–84.
[23] Rasmussen, B., Borch, K., and Stärk, K. D., 2001. "Functional modelling as basis for studying individual and organisational factors–application to risk analysis of salmonella in pork". *Food Control,* **12**(3), pp. 157–164.
[24] Papadopoulos, Y., and McDermid, J. A., 1999. "Hierarchically performed hazard origin and propagation studies". In International Conference on Computer Safety, Reliability, and Security, Springer, pp. 139–152.
[25] Nakao, H., Katahira, M., Miyamoto, Y., and Leveson, N., 2011. "Safety guided design of crew return vehicle in concept design phase using stamp/stpa". In Proc. of the 5: th IAASS Conference, Citeseer, pp. 497–501.
[26] Laracy, J. R., and Leveson, N. G., 2007. "Apply stamp to critical infrastructure protection". In Technologies for Homeland Security, 2007 IEEE Conference on, IEEE, pp. 215–220.
[27] Dulac, N., and Leveson, N., 2004. "An approach to design for safety in complex systems". In Int. Symposium on Systems Engineering (INCOSE).
[28] Ishimatsu, T., Leveson, N. G., Thomas, J. P., Fleming, C. H., Katahira, M., Miyamoto, Y., Ujiie, R., Nakao, H., and Hoshino, N., 2014. "Hazard analysis of complex spacecraft using systems-theoretic process analysis". *Journal of Spacecraft and Rockets,* **51**(2), pp. 509–522.
[29] Jensen, D., Tumer, I. Y., and Kurtoglu, T., 2009. "Design of an electrical power system using a functional failure and flow state logic reasoning methodology". *Prognostics and Health Management Society*.
[30] Coatanéa, E., Nonsiri, S., Ritola, T., Tumer, I. Y., and Jensen, D. C., 2011. "A framework for building dimensionless behavioral models to aid in function-based failure propagation analysis". *Journal of Mechanical Design,* **133**(12), p. 121001.
[31] Papakonstantinou, N., Sierla, S., Jensen, D. C., and Tumer, I. Y., 2011. "Capturing interactions and emergent failure behavior in complex engineered systems at multiple scales". In ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. 1045–1054.
[32] Sierla, S., Tumer, I., Papakonstantinou, N., Koskinen, K., and Jensen, D., 2012. "Early integration of safety to the mechatronic system design process by the functional failure identification and propagation framework". *Mechatronics,* **22**(2), pp. 137–151.
[33] McIntire, M. G., Keshavarzi, E., Tumer, I. Y., and Hoyle, C., 2016. "Functional Models With Inherent Behavior: Towards a Framework for Safety Analysis Early in the Design of Complex Systems". In ASME 2016 International Mechanical Engineering Congress and Exposition, American Society of Mechanical Engineers, pp. V011T15A035–V011T15A035.
[34] Li, Z. S., and Mobin, M. S., 2015. "System reliability assessment incorporating interface and function failure". In Reliability and Maintainability Symposium (RAMS), 2015 Annual, IEEE, pp. 1–8.
[35] Oh, Y., Yoo, J., Cha, S., and Son, H. S., 2005. "Software safety analysis of function block diagrams using fault trees". *Reliability Engineering & System Safety,* **88**(3), pp. 215–228.
[36] Meshkat, L., Jenkins, S., Mandutianu, S., and Heron, V., 2008. "Automated generation of risk and failure models during early phase design". In Aerospace Conference, 2008 IEEE, IEEE, pp. 1–12.
[37] Krus, D., and Lough, K. G., 2009. "Function-based failure propagation for conceptual design". *AI EDAM,* **23**(4), pp. 409–426.
[38] Keshavarzi, E., McIntire, M., Goebel, K., Tumer, I. Y., and Hoyle, C., 2017. "Resilient System Design Using Cost-Risk Analysis With Functional Models". In ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, p. V02AT03A043.

[39] Keshavarzi, E., 2018. "Resilient design for complex engineered systems in the early design phase".
[40] Short, A.-R., Lai, A. D., and Van Bossuyt, D. L., 2017. "Conceptual design of sacrificial sub-systems: failure flow decision functions". *Research in Engineering Design*, pp. 1–16.
[41] Papadopoulos, Y., Walker, M., Parker, D., Rüde, E., Hamann, R., Uhlig, A., Grätz, U., and Lien, R., 2011. "Engineering failure analysis and design optimisation with hip-hops". *Engineering Failure Analysis,* **18**(2), pp. 590–608.
[42] Adachi, M., Papadopoulos, Y., Sharvia, S., Parker, D., and Tohdo, T., 2011. "An approach to optimization of fault tolerant architectures using hip-hops". *Software: Practice and Experience,* **41**(11), pp. 1303–1327.
[43] Mehr, A. F., and Tumer, I. Y., 2006. "Risk-based decision-making for managing resources during the design of complex space exploration systems". *Journal of Mechanical Design,* **128**(4), pp. 1014–1022.
[44] Hoyle, C., Tumer, I. Y., Mehr, A. F., and Chen, W., 2009. "Health management allocation during conceptual system design". *Journal of Computing and Information Science in Engineering,* **9**(2), p. 021002.
[45] Pahl, G., and Beitz, W., 2007. *Engineering design: a systematic approach*. Springer Science & Business Media.
[46] Hulse, D., Hoyle, C., Goebel, K., and Tumer, I., 2018. "Optimizing function-based fault propagation model resilience using expected cost scoring". In ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.
[47] Erden, M. S., Komoto, H., van Beek, T. J., D'Amelio, V., Echavarria, E., and Tomiyama, T., 2008. "A review of function modeling: Approaches and applications". *AI EDAM,* **22**(2), pp. 147–169.
[48] Stone, R. B., and Wood, K. L., 2000. "Development of a functional basis for design". *Journal of Mechanical Design,* **122**(4), pp. 359–370.
[49] Kruse, B., Gilz, T., Shea, K., and Eigner, M., 2014. "Systematic comparison of functional models in sysml for design library evaluation". *Procedia CIRP,* **21**, pp. 34–39.
[50] Ullman, D., 2009. *The mechanical design process*. McGraw-Hill Science/Engineering/Math.
[51] Ulrich, Karl Tand Eppinger, S., 2012. *Product design and development*. McGraw-Hill Education.
[52] Wood, K. L., Stone, R. B., Mcadams, D., Hirtz, J., and Szykman, S., 2002. A functional basis for engineering design: Reconciling and evolving previous efforts. Tech. rep., National Institute of Standards and Technology.
[53] Jänsch, J., and Birkhofer, H., 2006. "The development of the guideline vdi 2221-the change of direction". In DS 36: Proceedings DESIGN 2006, the 9th International Design Conference, Dubrovnik, Croatia.
[54] Holling, C. S., 1973. "Resilience and stability of ecological systems". *Annual review of ecology and systematics,* **4**(1), pp. 1–23.
[55] Holling, C. S., 1996. "Engineering resilience versus ecological resilience". *Engineering within ecological constraints,* **31**(1996), p. 32.
[56] Pimm, S. L., 1984. "The complexity and stability of ecosystems". *Nature,* **307**(5949), p. 321.
[57] Masten, A. S., 2001. "Ordinary magic: Resilience processes in development.". *American psychologist,* **56**(3), p. 227.
[58] Luthar, S. S., Cicchetti, D., and Becker, B., 2000. "The construct of resilience: A critical evaluation and guidelines for future work". *Child development,* **71**(3), pp. 543–562.
[59] Briguglio, L., Cordina, G., Farrugia, N., and Vella, S., 2009. "Economic vulnerability and resilience: concepts and measurements". *Oxford development studies,* **37**(3), pp. 229–247.
[60] Perrings, C., 2006. "Resilience and sustainable development". *Environment and Development Economics,* **11**(4), pp. 417–427.
[61] Saint-Arnaud, S., and Bernard, P., 2003. "Convergence or resilience? a hierarchical cluster analysis of the welfare regimes in advanced countries". *Current sociology,* **51**(5), pp. 499–527.
[62] Cohen, R., Erez, K., Ben-Avraham, D., and Havlin, S., 2000. "Resilience of the internet to random breakdowns". *Physical review letters,* **85**(21), p. 4626.
[63] Ash, J., and Newth, D., 2007. "Optimizing complex networks for resilience against cascading failure". *Physica A: Statistical Mechanics and its Applications,* **380**, pp. 673–683.
[64] Sterbenz, J. P., Hutchison, D., Çetinkaya, E. K., Jabbar, A., Rohrer, J. P., Schöller, M., and Smith, P., 2010. "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines". *Computer Networks,* **54**(8), pp. 1245–1265.
[65] Lengnick-Hall, C. A., Beck, T. E., and Lengnick-Hall, M. L., 2011. "Developing a capacity for organizational resilience through strategic human resource management". *Human Resource Management Review,* **21**(3), pp. 243–255.
[66] Ponomarov, S. Y., and Holcomb, M. C., 2009. "Understanding the concept of supply chain resilience". *The International Journal of Logistics Management,* **20**(1), pp. 124–143.
[67] Chen, X., Xi, Z., and Jing, P., 2017. "A unified framework for evaluating supply chain reliability and resilience". *IEEE Transactions on Reliability,* **66**(4), pp. 1144–1156.
[68] Linkov, I., Bridges, T., Creutzig, F., Decker, J., Fox-Lent, C., Kröger, W., Lambert, J. H., Levermann, A., Montreuil, B., Nathwani, J., et al., 2014. "Changing the resilience paradigm". *Nature Climate Change,* **4**(6), p. 407.
[69] Hosseini, S., Barker, K., and Ramirez-Marquez, J. E., 2016. "A review of definitions and measures of system resilience". *Reliability Engineering & System Safety,* **145**, pp. 47–61.

[70] Yodo, N., and Wang, P., 2016. "Engineering resilience quantification and system design implications: A literature survey". *Journal of Mechanical Design,* **138**(11), p. 111408.
[71] Li, J., and Xi, Z., 2014. "Engineering recoverability: A new indicator of design for engineering resilience". In ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. V02AT03A044–V02AT03A044.
[72] Hazelrigg, G. A., 1998. "A framework for decision-based engineering design". *Journal of Mechanical Design,* **120**(4), pp. 653–658.
[73] Von Neumann, J., and Morgenstern, O., 2007. *Theory of games and economic behavior (commemorative edition)*. Princeton university press.
[74] Thurston, D. L., 2006. "Utility function fundamentals". In *Decision making in engineering design*. ASME Press.
[75] Hazelrigg, G. A., 1999. "An axiomatic framework for engineering design". *Journal of Mechanical Design,* **121**(3), pp. 342–347.
[76] Gu, X., Renaud, J. E., Ashe, L. M., Batill, S. M., Budhiraja, A. S., and Krajewski, L. J., 2002. "Decision-based collaborative optimization". *Journal of Mechanical Design,* **124**(1), pp. 1–13.
[77] Wassenaar, H. J., and Chen, W., 2003. "An approach to decision-based design with discrete choice analysis for demand modeling". *Journal of Mechanical Design,* **125**(3), pp. 490–497.
[78] Collopy, P. D., and Hollingsworth, P. M., 2011. "Value-driven design". *Journal of Aircraft,* **48**(3), pp. 749–759.
[79] Kmenta, S., and Ishii, K., 2000. "Scenario-based fmea: a life cycle cost perspective". In Proc. ASME Design Engineering Technical Conf. Baltimore, MD.
[80] Hu, C., and MacKenzie, C. A., 2017. "Optimizing resilience when designing engineered systems". In ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. V02AT03A047–V02AT03A047.
[81] Haimes, Y. Y., 2009. "On the definition of resilience in systems". *Risk Analysis,* **29**(4), pp. 498–501.
[82] Henry, D., and Ramirez-Marquez, J. E., 2012. "Generic metrics and quantitative approaches for system resilience as a function of time". *Reliability Engineering & System Safety,* **99**, pp. 114–122.
[83] Hulse, Daniel; Tumer, K. H. C. T. I., 2018. "Modeling multidisciplinary design with multiagent learning". *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*.
[84] Helms, B., Shea, K., and Hoisl, F., 2009. "A framework for computational design synthesis based on graph-grammars and function-behavior-structure". In ASME 2009 international design engineering technical conferences and computers and information in engineering conference, American Society of Mechanical Engineers, pp. 841–851.
[85] Sridharan, P., and Campbell, M. I., 2004. "A grammar for function structures". In ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. 41–55.
[86] Martins, J. R., and Lambe, A. B., 2013. "Multidisciplinary design optimization: a survey of architectures". *AIAA journal,* **51**(9), pp. 2049–2075.
[87] Stamatelatos, M., Vesely, W., Dugan, J., Fragola, J., Minarick, J., and Railsback, J., 2002. "Fault tree handbook with aerospace applications".
[88] Keshavarzi, E., McIntire, M., and Hoyle, C., 2015. "Dynamic design using the kalman filter for flexible systems with epistemic uncertainty". In ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. V02BT03A019–V02BT03A019.

Appendices

Appendix A: Monopropellant System Variants

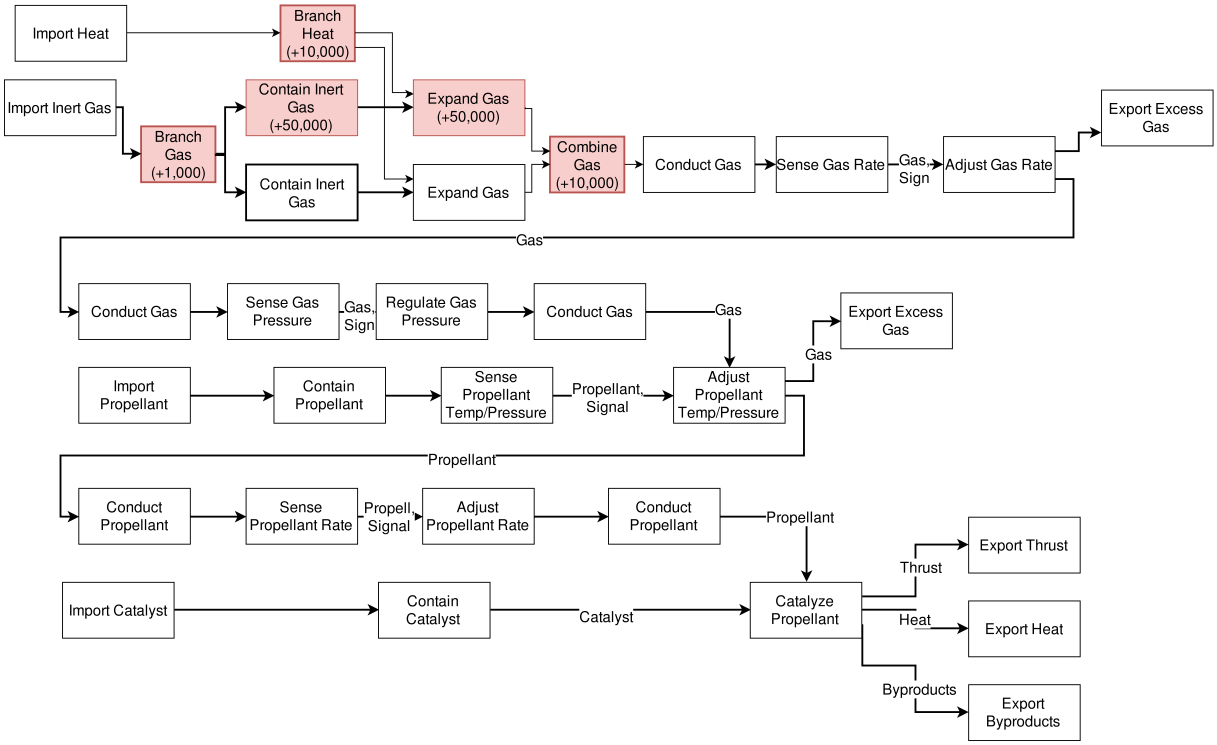

Fig. 10. Design Variant 1: Redundant Gas Tanks.

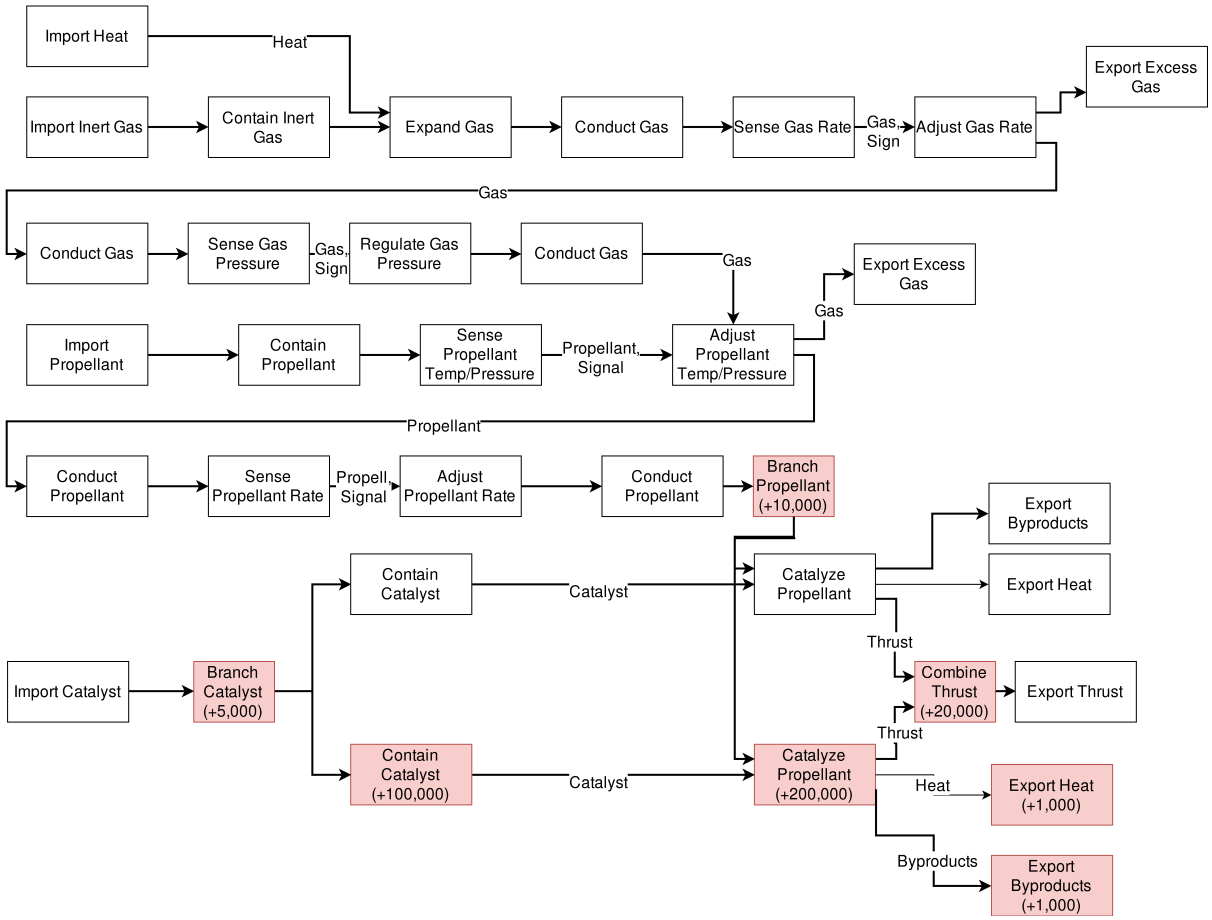

Fig. 11. Design Variant 2: Redundant Thrusters.

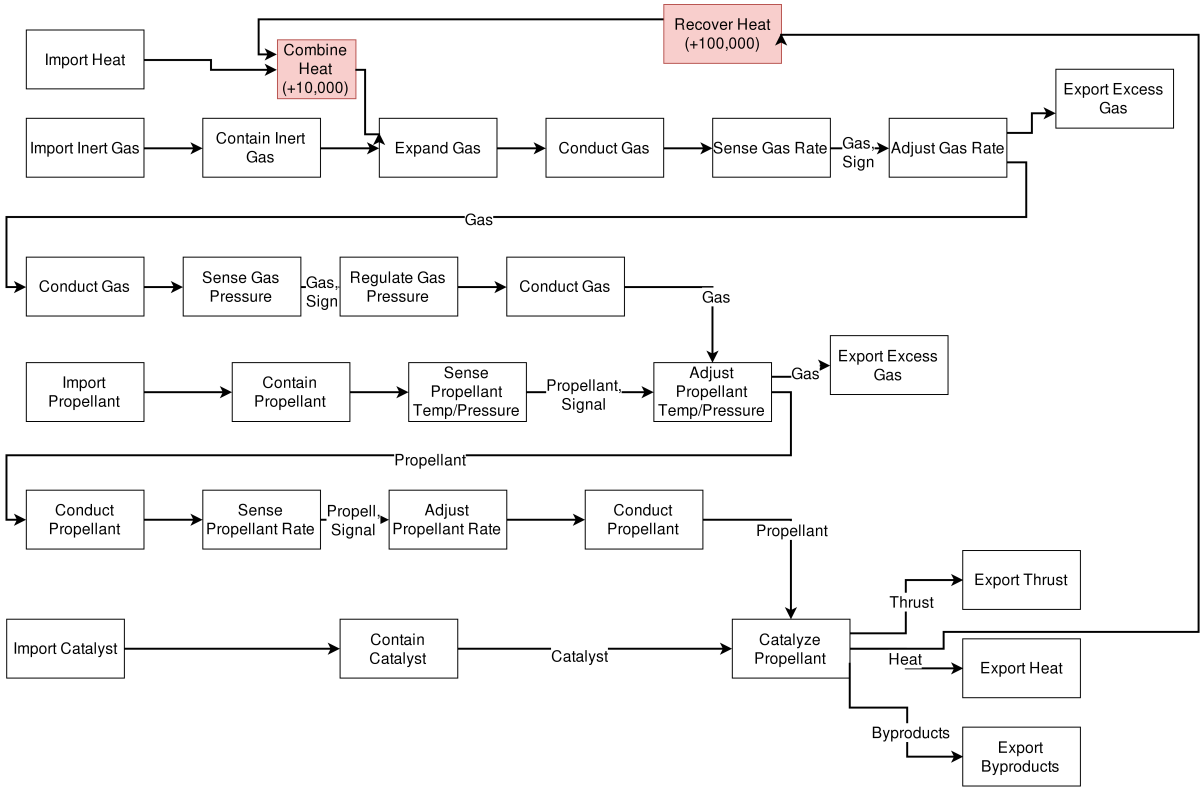

Fig. 12. Design Variant 3: Heat Recovery System.

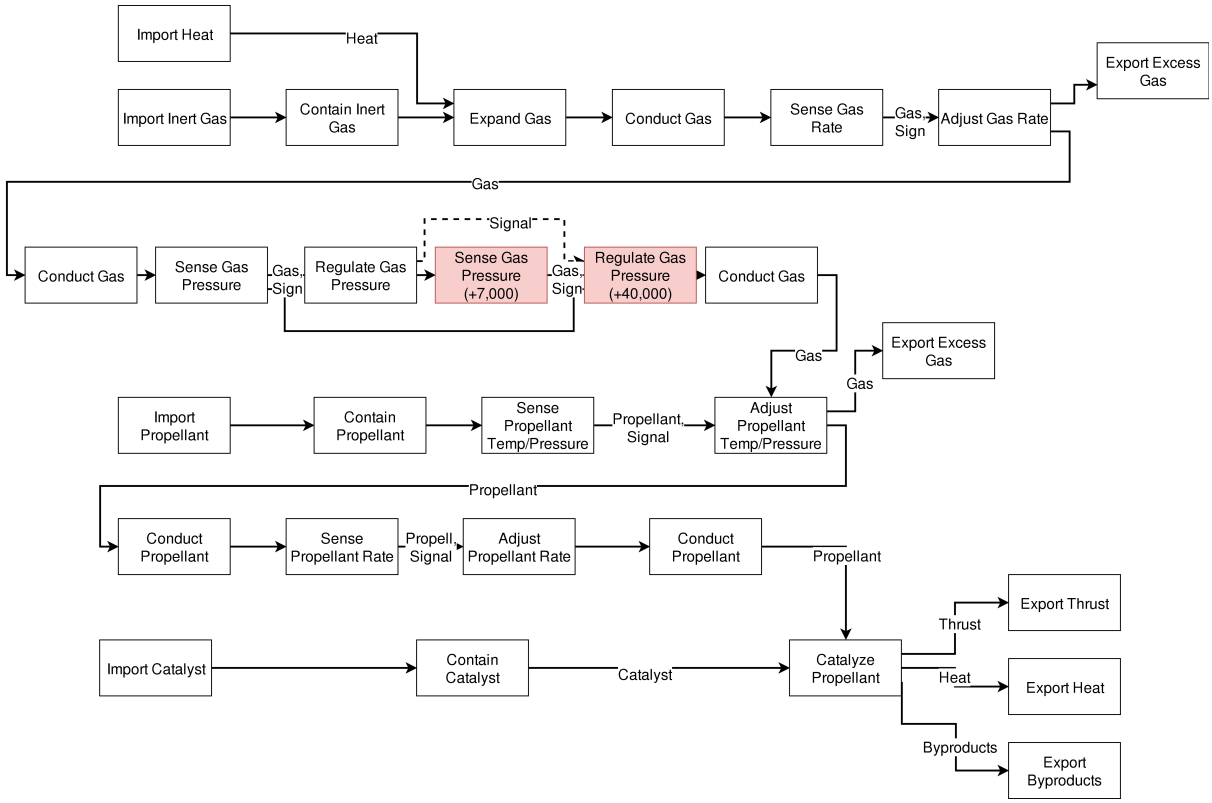

Fig. 13. Design Variant 4: Redundant Pressure Regulators.

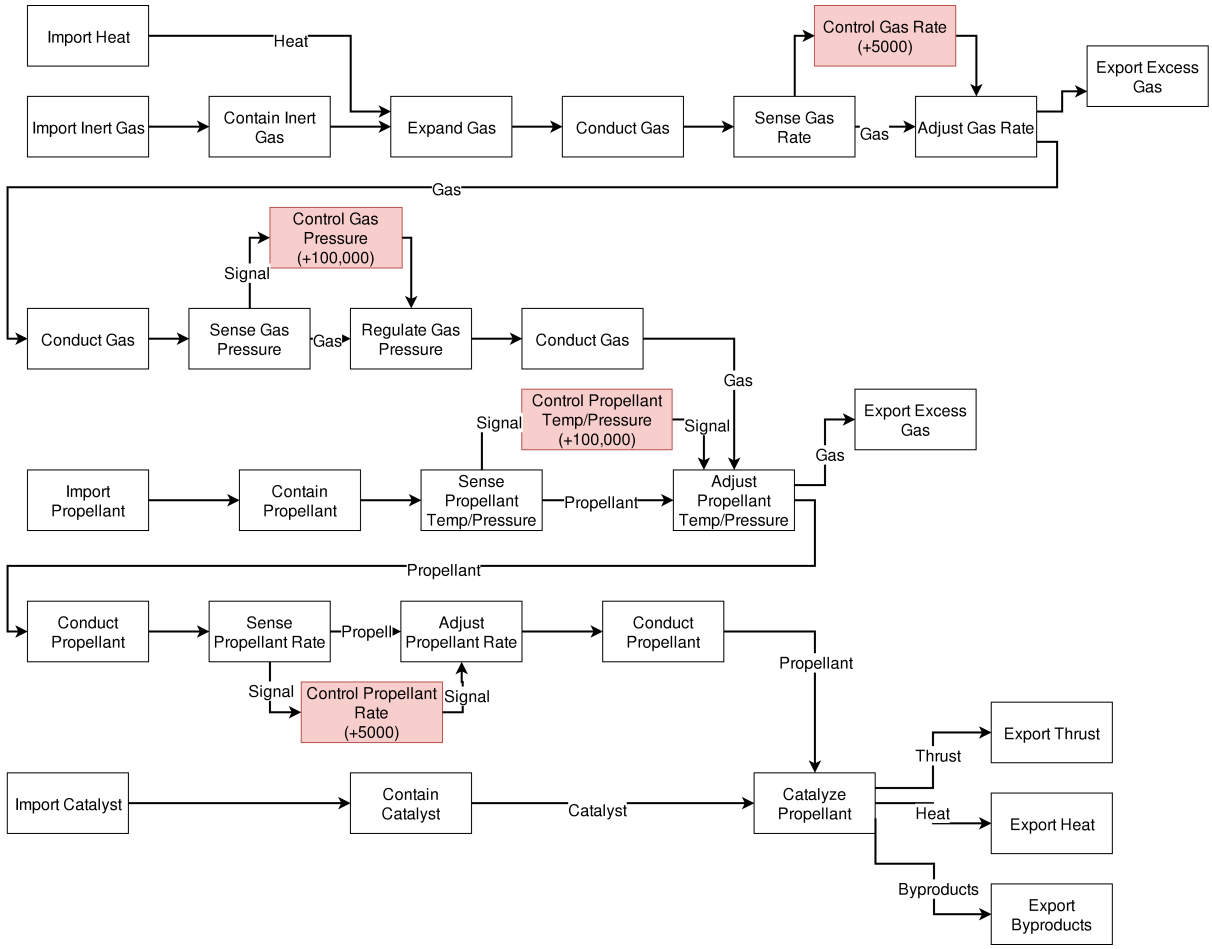

Fig. 14. Design Variant 5: Optimized Control Features.

Table 3. Design and operational changes which may be considered to optimize the adapted RISCS scoring, their related costs, difficulties to optimization, and value.

Design Change Type	Design Change	Difficulties	Value
Structural	New Structure	Determining context and purpose, necessary functional relationships, behavior of functions in new contexts.	Most impact and ability to explore novel solutions.
	Redundancy [38]	Predicting the effect of potential performance couplings on operational costs.	Reduces impact of individual failures. Enables behavioral consideration of entire redundant function chains, rather than those internal to a function.
	Function Order [38]	Predicting how function order might change design and operating costs. Determining behavioral impact of different function order.	Potentially inexpensive solution to lowering risk or reducing failure effects.
	Routing alternative flow paths (e.g. unused waste flows as inputs) [38]	Determining if flow path's effect on function behavior given a function's flow input requirements.	Ability to create resilience with less inherent cost increase than in other strategies (e.g. redundancy, excess capacity).
Functional Parameters	Redundancy	Predicting effect of potential performance couplings.	Easy to model and optimize. Enables consideration of redundancy without changing model structure.
	Assumed Realization/ Function Resources [43]	Potential internal and external compatibility couplings	Ability to represent tradeoff between cost and quality (mode probabilities and costs as well as function costs).
	Function Modes	Couplings with assumed realization.	Ability to represent differences in behaviors of functions.
	Conditional Logic [40]	Predicting design cost of flexibility required to allow different decisions to be made.	Enables representing the response of control systems and built-in robustness compensating for failures as well as sacrificial subsystems, etc.
Operational	Modes to recover	Computational expense in determining recovery in every scenario.	Ability to represent resilient operational decision-making and repair.
	Maintenance and Health Management [38] [44]	Determining effect on failure probability given time representation. Difficult to model with fault propagation.	Ability to represent operational ability to lower fault probability.

Table 4. Cost flow state matrix for the monopropellant thrust function, in billions.

Effort Health	Rate Health				
	Zero	Low	Nominal	High	Highest
Zero	4.5	4	3.5	4.25	5
Low	4	2.5	1.0	0.75	5
Nominal	3.5	1.0	0	1.0	5
High	4.25	0.5	1.0	2.5	5.5
Highest	5	5	5	5.5	5.5

Table 5. Generated Monopropellant Designs over Different Mission Utilities.

	Ctr1 Low	Ctrl1 High	Ctrl2 Low	Ctrl2 High	Ctrl3 Low	Ctrl3 High	Ctrl4 Low	Ctrl4 High
Feature Used	0	0	1	1	1	1	0	1
Feature Cost	550000	5000	50000	50000	50000	50000	2050000	5000