2019

Toward a functional failure analysis method of identifying and mitigating spurious system emissions in a system of systems

Van Bossuyt, Douglas L.; Arlitt, Ryan M.

ASME

Van Bossuyt, Douglas L., and Ryan M. Arlitt. "Toward a functional failure analysis method of identifying and mitigating spurious system emissions in a system of systems." International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Vol. 59179. American Society of Mechanical Engineers, 2019.

http://hdl.handle.net/10945/65179

Proceedings of the ASME 2019
International Design Engineering Technical Conferences
and Computers and Information in Engineering Conference
IDETC/CIE2019
August 18-21, 2019, Anaheim, CA, USA

DETC2019-98255

TOWARD A FUNCTIONAL FAILURE ANALYSIS METHOD OF IDENTIFYING AND MITIGATING SPURIOUS SYSTEM EMISSIONS IN A SYSTEM OF SYSTEMS

Douglas L. Van Bossuyt[*]
Systems Engineering Department
Naval Postgraduate School
Monterey, California 93943
Email: douglas.vanbossuyt@nps.edu

Ryan M. Arlitt
Department of Mechanical Engineering
Technical University of Denmark
DK- 2800 Kgs. Lyngby
Denmark
Email: rmarl@mek.dtu.dk

ABSTRACT

Increasingly tight coupling and heavy connectedness in systems of systems (SoS) presents new problems for systems designers and engineers. While the failure of one system within a SoS may produce little collateral damage beyond a loss in SoS capability, a highly interconnected SoS can experience significant damage when one member system fails in an unanticipated way. It is therefore important to develop systems that are ”good neighbors” with the other systems in a SoS by failing in ways that do not further degrade a SoS’s ability to complete its mission.

This paper presents a method to (1) analyze a system for potential spurious emissions and (2) choose mitigation strategies that provide the best return on investment for the SoS. The method is suited for use during the system architecture phase of the system design process. A functional and flow approach to analyzing spurious emissions and developing mitigation strategies is used in the method. Use of the method may result in a system that causes less SoS damage during a failure event.

INTRODUCTION

As the field of system of systems (SoS) engineering has developed over the last several years, an emerging area of interest is how well member systems behave with other systems. Many systems architects and systems engineers desire systems that are ”good neighbors” to other systems within the SoS in both nominal operation and in degraded or failed system states. While industry-standard failure modes and effects analysis (FMEA) and probabilistic risk assessment (PRA) techniques are currently being used to help design systems that are ”good neighbors,” there is a need for a more nuanced and robust approach to analyzing the effects that one system operating in a degraded or failed state has on its SoS neighbors.

We care as system designers and engineers to develop systems that do not damage systems around them when they fail because preventing such damaging events helps to ensure a SoS achieves its goals and mission objectives. This is becoming especially important as SoS become more tightly connected, such as the growth in the Internet of Things (IoT) and in digital interoperability initiatives within the Department of Defense (DoD). The systems that we design need to fail gracefully and contain their failure events to minimize damage to the SoS. For instance, enclosing turbo-machinery in shrouds to prevent shrapnel from a rapid unexpected fan disk deconstruction event spraying across a factory floor is a common-sense measure that can save significant damage to other equipment and potentially save the lives of nearby personnel.

SPECIFIC CONTRIBUTION

In this paper, we present a conceptual system design method – appropriate for the early system architecture phase of the systems engineering process – to identify and mitigate potential spurious system emissions in the form of failure flows. The method

[*] Address all correspondence to this author.

uses functional modeling and functional abstraction, and probabilistic failure analysis techniques to identify potential spurious failure flow emissions. In this context, a spurious failure flow emission is a flow emitted by a function that (1) is not present in nominal operation and (2) leaves the system boundary. The potential spurious emissions are identified on a per-function basis and with special attention paid to potential low probability but high consequence events. The method further balances system requirements and constraints with implementing mitigation strategies in order to protect the large system of systems.

The method presented here specifically is intended to be used before large system architectural decisions have been made and when there is still significant flexibility in the fundamental functional design of the system. Other methods that examine spurious failure flow emissions generally do so after functional architectures have been solidified and component solutions to functions have been identified. Mitigation strategies that result from analysis performed later in the systems engineering and design processes often result in mitigation subsystems that are afterthoughts. We instead present a method that allows spurious system emissions mitigation strategies to be integrated during and implemented from the very earliest functional modeling efforts of a new system that is to join a SoS.

The method is intended to complement existing failure analysis techniques discussed in subsequent sections. However, it is not intended to be a direct add-on to existing techniques such as FMEA or PRA. The method most closely integrates with other existing functional failure analysis techniques.

BACKGROUND AND RELATED WORK

Several areas of related research and professional practice are of importance to the research presented in this paper. This section reviews relevant literature on the systems engineering and architecture process, functional modeling, failure and risk analysis methods, and methods related to the research presented here.

SYSTEMS ENGINEERING AND ARCHITECTURE

The systems engineering process describes how complex systems are brought from initial concept to production, customer delivery and use, maintenance and upgrade, and disposal [1, 2]. Groups such as the International Council on Systems Engineering (INCOSE) provide relevant detailed and specific knowledge on the process [3]. Of particular interest to this research is the early phase of systems engineering that is encompassed by the system architecting process [2, 4, 5]. System architecting includes developing customer needs statements, design reference missions, system requirements, functional system models, trade-off studies, and a variety of other work products [2,5,6]. This information is generally stored in a database that can be interpreted by a system architecture framework [7] such as the Department of Defense Architecture Framework (DoDAF) [8] which was specifically modified in the V2.0 release to heavily encourage the use of a database [9]. Many other frameworks are available that are suitable for a variety of industries and products [7,10–13]. Of particular interest to this research is the functional modeling and trade-off studies conducted during the system architecture phase of systems engineering.

Another aspect of systems engineering that is important to this research is the concept of a SoS. SoS is an emerging area of interest in systems engineering that focuses on understanding how groups of systems interact and work together [14–17]. In recent years, advances have been made in understanding how to develop SoS that are mixed legacy and new development systems, that are rapidly assembled for a particular SoS goal or mission, and a host of other advances [16, 18–22]. This research is specifically interested in how the risk of SoS failure is understood and SoS failure risk is mitigated [23].

While some systems engineering and architecture methods do suggest that spurious system emissions be analyzed and mitigated, such efforts are only implicitly indicated in the system architecture (conceptual design) phase through requirements development and management. Many standards (DoD, ISO, FCC, etc.) explicitly call for requirements on spurious system emissions. However, efforts to determine potential spurious system emissions generally fall much later in the system design process and sometimes do not occur until physical prototype testing occurs.

FUNCTIONAL MODELING

Functional modeling is a method of modeling how a system works at the fundamental functional level [24] and is a subset of system modeling [25–28]. There are a variety of different taxonomies available to produce functional models [29]. We prefer the Functional Basis for Engineering Design (FBED) and use it throughout this paper [30, 31]. However, other functional modeling taxonomies [32, 33] may be more appropriate for specific applications.

A functional modeling taxonomy generally is composed of functions and flows. Functions act upon the flows to transform incoming flows to different outgoing flows. For example, a function may accept electrical energy in and output thermal energy [30]. Functions and flows are connected to their physical component solutions through databases and repositories [34–39]. One function may have many potential component solutions. For example, a convert electrical energy to rotational energy function may be satisfied by several types of electrical motors. Similarly, one flow may have many different physical manifestations, such as an energy-electrical flow being physically manifested as an alternating current or a direct current [40].

A variety of techniques to analyze a risk, sustainability, failure, and other useful system design concerns have been developed around functional modeling. Topics important to the method presented in this paper are discussed in a subsequent subsection. However, no functional modeling-based technique that we are aware of focuses on mitigating spurious system emissions.

FAILURE ANALYSIS

Failure analysis has become an integral part of systems engineering over the past 75 years. A major milestone in the advancement of understanding and preventing failure in system design was the advent of Failure Modes and Effects Analysis (FMEA) [41] and the related Failure Modes Effects and Criticality Analysis (FMECA) [42, 43]. Both are extensively used throughout the DoD and DoD contractors [41] (where MIL-STD-882E prescribes FMECA [44]), the automotive industry [45], and other industries [46].

PRA was developed in part as an effort to understand complex system failures and has roots in the nuclear power and aerospace industries [47–51]. PRA helped to popularize the concept of an initiating event – an event that is the starting point for a failure that propagates through a system [52–54]. Many systems engineered using PRA have the ability to catch incipient failures and either transition to a safe shutdown state or continue operating either nominally or in a degraded state while repairs are made [55–57]. PRA produces cut-sets which are sets of events that must all occur in order for a system failure to occur. Each cut-set represents a specific sequence of events that leads to a system failure, and each cut-set is quantified with a probability of occurrence computed from the probability of each event in the chain of events [58, 59]. The probabilities of all cut-sets for a specific system can be summed to find the overall probability of a system failing [60]. This research uses the concept of initiating events, probabilistic mathematics, and cut-sets, all of which were largely developed by PRA.

Existing failure analysis methods do have capabilities to investigate and propose mitigation strategies for spurious system emissions. However, such work is generally done long after functional models of the systems have been completed and the conceptual design phase has ended. This constrains systems engineers and designers from making large architectural changes to address spurious systems emissions without incurring significant financial and schedule costs.

0.1 RESILIENCE, ROBUSTNESS, AND SURVIVABILITY ANALYSIS

Many methods exist to examine resilience [61–63], robustness [64, 65], and survivability [66, 67] of systems both from the standpoint of challenges introduced from the environment and from the standpoint of other systems within a SoS. The topics of resilience, robustness, and survivability have significant overlap. We suggest that the high-level goal of each of the topics is the same: the system must work in spite of non-nominal and potentially unexpected conditions encountered.

Techniques to examine resilience, robustness, and survivability are found throughout much of the system design process. However, none that we are aware of take a quantitative functional approach.

FUNCTIONAL FAILURE ANALYSIS

Reliability block diagrams (RBDs) were developed in part to better understand how system reliability is impacted by the way functions are connected together in a system [68, 69]. Some implementations of RBDs use functional representations of systems while other RBD implementations use components and subsystem representations [70–73]. In order to improve system reliability, a common technique is to have multiple parallel redundant critical functions [24]. RBDs are used extensively in industries including defense [74], aerospace [75], and more [76, 77].

A family of methods has been developed over the past two decades to analyze failure events from a functional perspective. Initial work was done to combine FMEAs and functional models [78] which was followed by other early techniques to analyze risk of failure from a functional perspective [79–87]. Work over the last decade has focused on a family of methods based around the function failure identification and propagation (FFIP) method [88] and the companion flow state logic (FSL) method [89]. The FFIP family of methods has been expanded to examine how prognostics and health management systems can be designed during system architecture [90], how failure flows may jump between systems using the uncoupled failure flow state reasoner (UFFSR) [91], how to protect against uncoupled failure flows within a system [92], how systems can deal with a variety of unanticipated external initiating events in a SoS [23], and several other important advances [93–100]. We use the FFIP family of methods extensively throughout the research in this paper.

While one functional failure method in particular that is discussed in the following section does focus on spurious system emissions, it only looks at how to protect against receiving such emissions rather than not emitting in the first place. This is in contrast with the method that we present below which specifically focuses on being a ”good neighbor” and not allowing spurious system emissions occur.

DIRECTLY APPLICABLE RELATED WORK

Several related works are of relevance to this paper. For instance, failures jumping between unrelated and unconnected functions was first explored by O’Halloran et. al. through the UFFSR method [91]. Later work expanded the idea further to include initiating events entering systems along non-nominal flow paths [23]. We extend the idea in this paper to include failure flows exiting systems along non-nominal flow paths.

The irrational system behavior analysis method [23] examines how systems can respond to unexpected external initiating events in a SoS. Such initiating events are caused by a system within a SoS behaving in an unexpected manner. This research is in part an effort to address such initiating events from the perspective of the system emitting the unexpected failure flows that result in unexpected external initiating events for other systems.

The concept of mitigating failure flows through dedicated failure flow arrestor functions [92] was proposed in response to the UFFSR method [91] to protect critical functions within a system from failure flows jumping between unrelated subsystems. We expand upon the idea to develop mitigation strategies for failure flows that exit system boundaries as spurious emissions.

In summary: the research we present in this paper is based upon a significant heritage of research and professional practice in systems engineering and failure analysis. While the methods available to systems designers and engineers to understand failures, model systems, and mitigate potential spurious system emissions is large, no existing method combines all of these things during the conceptual system design phase where large architectural changes can be made with minimal time or cost penalty to a system design project. The method we present below fills a gap and a need in the existing literature and methodologies.

METHODOLOGY

In this section we present a method to identify potential spurious system emissions, quantify potential emission probabilities, identify potential mitigation strategies to prevent spurious emissions from causing harm to other systems, and conduct trade-off studies to determine the best course of action moving forward with the system design. The method is useful for understanding and improving the suitability of a system for inclusion in a SoS where it is important to be a good neighbor with the other systems. The system architecture phase is the intended place within the systems engineering process to use the method.

The below method is most appropriate for systems that have well-understood components where sufficient failure data is available. For instance, a new airplane design using components that are either commercial off the shelf (COTS) or are sufficiently similar to COTS to use COTS failure data is an acceptable system to be analyzed by the method. A system containing many novel components that have no failure data available and no COTS equivalents is not appropriate.

The method is intended to be semi-automated. While it is possible to conduct the method by hand, an unacceptable amount of time would be necessary to iterate through the method. Instead, the initial pass through the method is partially manually conducted to build the requisite models and databases. Subsequent iterations of the method are handled automatically but with human-in-the-loop analysis of results at the end of each iteration to ensure that the results make sense.

CASE STUDY

We introduce an illustrative case study of an autonomous vehicle that is being designed to enter service with an autonomous logistics system (the SoS). The SoS operates in a desert environment carrying materiel to a forward operating base. This frees up military personnel and contractors from routine and potentially dangerous resupply missions [101] to concentrate on other high value activities. There are other constituent members of the SoS including ground control stations, command and control relay stations, and other autonomous systems such as autonomous ground vehicles. The SoS is expected to have additional heterogeneous members added in the future as logistics demands dictate expansion to a nearby littoral zone where autonomous surface and underwater vessels may be used.

The system architecture process has already down-selected to the production of an unmanned aerial vehicle (UAV) for the specific payloads and mission constraints identified during the development of the customer needs statement, the design reference mission, and the system requirements (shown in Table 1). The UAV was down-selected from other potential options (e.g.: surface vehicles, ballistic trajectory delivery, etc.) because of the need for rapid payload delivery of cargo that is sensitive to shock and vibration.

Req #	Requirement
#1	Carry 10kg 5km
#2	Complete round-trip transit with 99% success rate
#3	Communicate with ground control station at 1.5Mbit/sec TX/RX
⋮	⋮

TABLE 1. Generic set of requirements for a UAV used to carry cargo.

While the system presented in this paper is generic, it takes inspiration from real systems. The importance of the case study is not to show a real system designed to mitigate spurious system emissions in a real SoS. The case study is used to illustrate the method presented below and has been intentionally simplified to better highlight the specific contributions of the method.

PREPARATORY STEP

Prior to using the method, several preparations must be made. They fall into two categories: 1) to prepare an FFIP model of the system that will be used throughout the method, and 2) to prepare information for the trade-off study conducted in Steps 6 and 7.

First, a functional model of the system of interest must be developed. In order to do this, a functional taxonomy must be chosen for use throughout the project. We recommend using the FBED taxonomy [30] although there are many other choices available to the practitioner. If following the common system architecture process, a practitioner likely already will have a functional model as part of modeling the system [5]. Figure 1 shows a high-level functional model for a UAV developed using FBED.

Following the development of a functional model of the system, a function to component relational database must be developed [37, 39]. This entails identifying potential component solutions to functions. Historical data about similar systems is useful to build out such databases. For instance, design repositories may hold such information [34]. Table 2 demonstrates what a function to component relational database contains.

Function	Component Solution
Channel-Guide-Rotate	DC Electrical Motor
	AC Electrical Motor
	Pneumatic Motor
Provision-Supply	Lithium Polymer Battery
	Supercapacitor
	Lead Acid Battery
⋮	⋮

TABLE 2. Generic function to component repository for a UAV.

Note that we have omitted cost and performance data from Table 2 and from the rest of the methodology for nominal functions. However in practice, this information is generally included in such databases and used to conduct trade-off studies when selecting specific components to solve functions [102, 103]. We have omitted this information here to focus on the contributions of this method.

The function to component relational database must then be expanded to include failure data and failure modes [89,104]. This includes historical component failure information that is then wrapped up to the functional level [102, 105]. This also includes information on how failure flows can be received into a function, how the function might be impacted by the failure flows, and what failure flows the function might emit [89,95,104]. This information is then placed into an algorithm for future use. Algorithm 1 demonstrates what this looks like for a generic function using a modified version of FSL [89] where more details on developing FSL pseudocode.

```
input  : Nominal Flows, Failure Flows, System Operation Status
output: Nominal Flows, Failure Flows
Let P1 = Sample from a Uniform(0,1) Random Distribution
Let P2 = Sample from a Uniform(0,1) Random Distribution

while System Operation Status == True do
    if Energy-Electrical Failure Import Flow == True then
        if $0.2 < P1 < 0.4$ then
            Export Failure Flow = Energy-Thermal, Material-Mixture-Gas/Solid Export Nominal Flow = False
        end
        else if $P1 \leq 0.2$) then
            Export Failure Flow = Energy-Electrical Export Nominal Flow = False
        end
        else
            if Electrical-Energy Nominal Import == True then
                Export Failure Flow = False Export Nominal Flow = Energy-Electrical
    if Material-Solid-Object Failure Import Flow == True then
        if $P2 < 0.2$ then
            Export Failure Flow = Energy-Thermal, Material-Mixture-Gas/Solid Export Nominal Flow = False
        end
        else
            Export Failure Flow = False
            if Electrical Energy Nominal Import == True then
                Export Nominal Flow = Energy-Electrical
            else
                Export Nominal Flow = False
            end
    if Failure Import Flow == False then
        if Electrical-Energy Nominal Import == True then
            Export Nominal Flow = Energy-Electrical
        else
            Export Nominal Flow = False
        end
    else
        Export Nominal Flow = False
    end
end
```

Algorithm 1: Failure flow information for a generic Provision-Supply function.

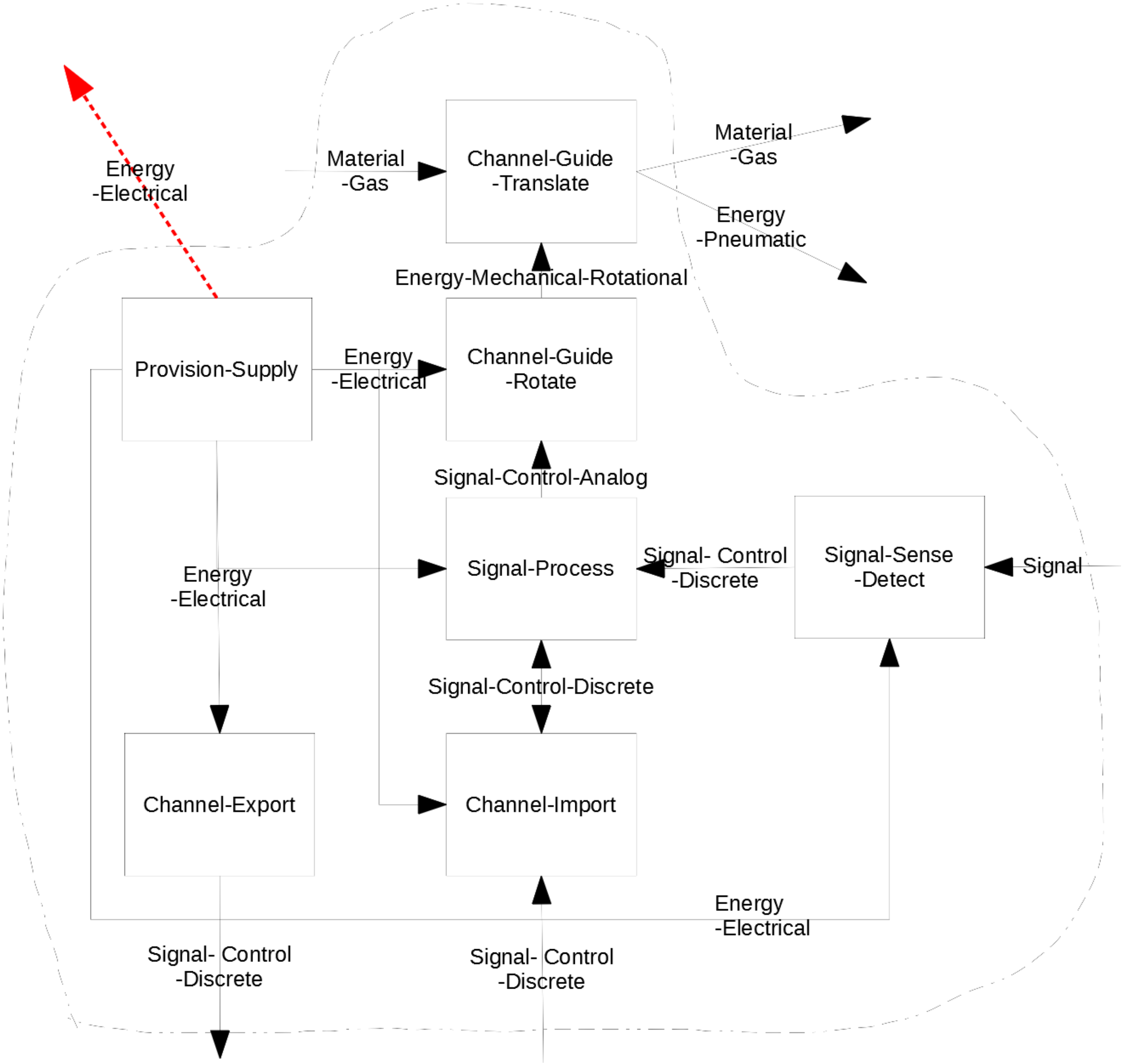

FIGURE 1. Generic functional model of a UAV developed using the FBED taxonomy. Note the dotted and dashed line represents the system boundary. The dashed line represents a failure flow system emission.

Next, system requirements information must be collected. Of specific interest is performance metrics and system constraints. Cost and failure probability constraints in particular are required to use this method. Other requirements and constraints will vary depending upon the individual system. Both the systems engineering process and the mechanical design process advocate for developing system or product requirements [5, 106] at the start of their respective development cycles. Thus, this method assumes that adequate requirements have already been developed for the system. Table 1 shows a generic set of requirements for a UAV. This information will help to set goals for the trade-off studies conducted in Step 6 of the method.

Finally, an analysis of the system of interest's place in a larger SOS environment must be undertaken. Questions to ask are: (1) what other systems are present, (2) how important is it that each system continues to function, (3) what is the cost of having a system fail, and (4) what external event(s) may cause a system to fail [23]. The resulting information then must be placed in terms of consequences for other systems within the SOS failing. An example of consequence data is shown in Table 3 for a generic heterogeneous mixed UAV and unmanned ground vehicle (UGV) swarm. The consequence is determined by the cost distribution function, C_e, defined as the probability density of the cost of a system damaging other systems within a SoS from emitted failure flows.

STEP 1: ANALYSIS OF EACH FUNCTION AND WHAT IT CONCEIVABLY COULD EMIT

In the preparatory step, a function to component relational database was developed. In addition to function to component mapping and related information [37, 39], the previously developed database contains failure modes information [80] as shown in Algorithm 1. Now the failure modes must be expanded to go beyond failures that have been previously observed. Previous work has dealt with expanding knowledge of what failure flows may impact a function [23] where analysis of the entire flow set is conducted for each function. This analysis must be done for each function in order to build out a complete algorithm beyond what was done in the Preparatory Step.

We propose a goal of examining all possible failure flows that may be emitted by a function – similar in principle to the broad identification of failures via FMEA. This requires careful examination of what may be emitted by a function in all of its failure states. After understanding what failure flows might be received by a function, then a better understanding of what failure flows may be emitted by a function can be developed. We caution the practitioner that even the most outlandish scenarios for generating a specific failure flow emission from a function must be considered. While the fields of risk analysis and reliability engineering have gotten very good at identifying potential failures based on what has been observed in similar systems in the past, highly improbable failures that were discounted or not considered during the conceptual phase of systems engineering continue to happen [107, 108].

To identify a high proportion of all possible emitted failure flows, we propose working backwards from the flow taxonomy of FBED to attempt to disprove the hypothesis that each of the flow types can be emitted as a failure flow by the function in question. For instance, the Energy-Thermal flow may be generated by a function such as Control-Stop-Inhibit where the component solution is a metal barrier if the function receives a failure flow input such as Energy-Vibration where the flow's physical solution is a high frequency, high amplitude vibration caused by an unanticipated failure somewhere else in the system. Table 4 presents an example of a generic function where received failure flows are connected to emitted failure flows. The crossed-out failure flow exports represent those exports that have been found to be impossible to create regardless of the failure flow import to the function. The component solutions for the function examined in Table 4 indicate which component solutions, as seen in Table 2, may create which failure flows. Significant additional documentation is developed for each potential failure flow export to strongly disprove that it is possible. We recommend that this document take a form similar to workbooks developed for basic events and initiating events in PRA models in the nuclear power industry [109]. However, we acknowledge that this may be impractical for early conceptual system design trade-off studies in which case, we advise that the practitioner identifies an acceptable level of rigor and documentation to follow when analyzing each potential failure flow export. In either case, it is important to maintain the link between the functional model and the physical solutions of functions and flows. An added layer of complexity in this step of the method is that different physical solutions to a function may have different responses to failure flow imports.

The above newly identified failure flow exports from the function are then appended to the function's failure logic algorithm. We have omitted a demonstration of this part of the step due to space constraints.

STEP 2: EVALUATE ALL POTENTIAL FLOW PATHS THROUGH THE SYSTEM

Following the analysis of what each function may conceivably emit, the next step is to evaluate all potential failure flow paths through the system of interest. We recommend using established FFIP/FSL [89, 104] and UFFSR [91] methodology for determining what flows may traverse a system. Note however that at this point in time, we are not assigning probabilities to individual flow paths, functional failure events, or initiating events. The method intentionally does not assign probabilities at this step to avoid the pitfalls of truncation of failure flow paths that often occurs during FFIP-style analyses. While the PRA literature and other probabilistic approaches to failure analysis advises practitioners to set practical truncation limits, we suggest that the O'Halloran et. al. approach [97] be followed instead to find all potential failure flow paths. This helps to prevent missing potential low probability, high consequence failure flows. Further, certain very low probability failure flow paths that produce failure flow emissions contribute to a higher probability of the failure flow emission occurring.

The drawback to examining all possible failure flow paths

Failure Flow Exports from System of Interest that Lead to Initiating Events for Other Systems in SoS	Consequence	C_e
Energy-Electrical	Static-electric discharges during dust storms caused by UAV rotors or propellers can short out onboard electronics of nearby vehicles leading to loss of both UAVs and UGVs	\$5M
Material-Solid-Particulate	Large particulate from crashed UAVs can clog air vents and cause overheating of UGVs leading to disabled systems	\$1M
Material-Control-Analog	Interference with radio transceivers causes UAVs to automatically land regardless of terrain or of potential adversary presence	\$2M
⋮	⋮	⋮

TABLE 3. Consequence data for an mixed UAV and UGV SoS where consequence the cost distribution function C_e for the impact of the system on the SoS. In this example, each C_e is a point distribution.

through a system is that it becomes computationally expensive. O'Halloran et. al. [97] provides insights on how to overcome this issue. For instance, it may become necessary to truncate specific failure flow paths due to computational costs in which case, truncating based on the failure flow path length (i.e.: how many functions the failure flow traverses) may be appropriate.

Some example failure flow paths that exit the system boundary from the generic UAV system used throughout the method are presented in Table 5. Note that we are not assigning probabilities at this step in the method.

STEP 3: DETERMINE PROBABILITIES OF SPURIOUS EXITING FLOWS

After all failure flow paths have been identified, the next step is to quantify the probability of each failure flow emission from the system. This step diverges from established practices in the FFIP family of methods. Rather than stopping at producing cut-set results that are analyzed individually, the probability of each spurious failure flow emission is developed from aggregating cut-sets into groups based on the specific spurious failure flow emissions that they produce. In this context, cut-sets are defined as the path that each failure flow travels from the initial failure event to exiting the system as a spurious failure flow emission. The definition is in line with how cut-sets have been used in recent FFIP-related research [23, 95, 98, 99] and is similar to how cut-sets are defined in the PRA literature [60].

A table of system-level failure flows is generated from this step. Table 6 shows a representative set of data for a generic UAV system. Note that only the failure flow emissions identified through analysis of this method are present. Each failure flow emission type and probability of occurrence, PO_e, is listed where the probability is an aggregation of all failure flow path cut-sets that result in that particular failure flow emission type. PO_e can either be a point value or a probability density function although we recommend using probability density functions wherever possible. We recommend that all failure flow emissions that travel along both nominal flow export paths and along uncoupled flow export paths are represented in this table. Failure flow system emissions for the purposes of this method are defined as flow paths that leave the system boundary.

Note that the probabilities shown in Table 6 are aggregated from the component solution probabilities shown in Table 4. In other words, Table 6 shows the combined probabilities of each function's component solutions. As decisions are made on how to satisfy individual functions with component solutions during the systems engineering process, the probability statistics will change. Because of this, we recommend rerunning the methodology as major design decisions are made and components are selected.

STEP 4: ANALYZE RESULTS

Now that failure flow system emissions are identified and the probabilities of occurrence are calculated, analysis of the results can be conducted. There are two potential approaches that practitioners may use at this step including: (1) identifying the highest probability of occurrence failure flow system emissions and rank ordering them according to their probabilities, and (2) examining the other systems within the SoS that may be impacted by the failure flow system emissions. The first approach is the standard approach often taken in FMEA and other analyses [6]. We advocate that the second approach be taken and provide further details below.

Table 3 from the preparatory step of the method provides a means for evaluating how failure flow system emissions from the system of interest may impact other systems in the SoS. A mapping of failure flow system emissions to initiating events of

TABLE 4. Generic example of examining potential failure flows and determining if they can occur using the FBED flow set for a Channel-Guide-Rotate function in a UAV system. Failure flows generated by specific component solutions are indicated on the right hand side of the table.

Failure Flow Imports			→	Failure Flow Exports			
Primary	**Secondary**	**Tertiary**	→	**Primary**	**Secondary**	**Tertiary**	**Component Solution(s) to Function**
				Material	~~Human~~		
Energy	Mechanical	Translational	→		Gas		DC Motor
Energy	Electrical		→		Liquid		AC Motor
Energy	Mechanical	Translational	→		Solid	Object	DC Motor, AC Motor
Energy	Mechanical	Translational	→			Particulate	DC Motor, AC Motor, Pneumatic Motor
						~~Composite~~	
					~~Plasma~~		
					~~Mixture~~	~~Gas-gas~~	
						~~Liquid-liquid~~	
						~~Solid-solid~~	
						~~Solid-Liquid~~	
						~~Liquid-Gas~~	
						~~Solid-Gas~~	
						~~Solid-Liquid-Gas~~	
						~~Colloidal~~	
				Signal	Status	~~Auditory~~	
						~~Olfactory~~	
						~~Tactile~~	
						~~Taste~~	
Energy	Mechanical	Pneumatic	→			Visual	Pneumatic Motor
Energy	Electrical		→		Control	Analog	AC Motor
Energy	Electromagnetic	Solar	→	”	”	”	AC Motor
						~~Discrete~~	
				Energy	~~Human~~		
					~~Acoustic~~		
					~~Biological~~		
					~~Chemical~~		
					~~Electrical~~		
Energy	Electrical		→		Electromagnetic	Optical	DC Motor
						~~Solar~~	
					~~Hydraulic~~		
					~~Magnetic~~		
					~~Mechanical~~	~~Rotational~~	
						~~Translational~~	
						~~Pneumatic~~	
					~~Radioactive/Nuclear~~		
Energy	Radioactive/Nuclear		→		Thermal		AC Motor, DC Motor, Pneumatic Motor

Flow Path #	Failure Flow Path
#1	Energy-Electrical → Provision-Supply → Energy-Electrical → Channel-Export → Signal-Control-Discrete
#2	Material-Mixture-Gas/Solid → Channel-Guide-Translate → Material-Solid
#3	Provision-Supply → Energy-Electrical → Channel-Guide-Rotate → Energy-Thermal
⋮	⋮

TABLE 5. Example failure flow paths of a generic UAV system that exit the system boundary.

Failure Flow System Emission	PO_e
Energy, Mechanical, Translational	2.2E-4/year
Material, Gas	4.3E-3/year
Signal, Status, Visual	5.6E-3/year
Material, Solid, Particulate	1.9E-2/year
Material, Liquid	8.3E-3/year
⋮	⋮

TABLE 6. Probability of occurrence of a representative set of failure flow system emissions for a generic UAV system.

other systems with a probability of negative consequence (P_e) to the other systems can be produced. The probability of the negative consequence can take the form of a point value or a distribution and is dictated by the information that is available to the practitioner. Early in a system design process, only point value estimates may be available but as the system architecture effort moves forward, more detailed probabilistic information may be available. Note that P_e does not indicate how severe the consequences are to the other systems in the SoS. The result of this effort can provide a reasonable indication of priorities for what failure flow system emissions are most important. Table 7 provides an example of what such efforts may produce.

Priority	Failure Flow System Emission	P_e
#1	Signal,Status,Visual	2.1E-4/year
#2	Material,Liquid	8.3E-5/year
#3	Material,Solid,Particulate	1.9E-5/year
⋮	⋮	⋮

TABLE 7. Priority ranking based on probability of adversely impacting other systems in the SOS with a failure flow leaving the system boundary as a spurious emission (P_e). Note that P_e is not the same as the probability of occurrence in Table 6 – P_e is the probability of negative consequences on other systems in the SoS while probability of occurrence is of the failure flow being emitted from the system.

While the analysis approach shown in Table 7 is sufficient for early analysis of the system to determine priorities for addressing failure flow system emissions, a more nuanced understanding of the order of importance is desirable. Table 7 determines priority of addressing failure flow system emissions but does not provide an indication of how important one failure flow system emission is as compared to another. For instance, a high probability and thus a high priority flow from Table 7 may have a very minimal impact on other systems in the SoS while a lower probability and priority flow may have a much more significant impact on other SoS systems. In extreme cases, a very low probability and low priority failure flow system emission may in fact cause catastrophic loss of the entire SoS. An example of this is a drone in a SoS of many drones emitting a large electromagnetic frequency (EMF) burst after coming into contact with power lines which may interrupt command and control of other nearby drones, resulting in them immediately attempting to land regardless of terrain features below them. This in turn may result in the damage, destruction, or loss of many drones.

We propose Emission Priority Distribution (EPD) as a metric to make comparisons between failure flow system emissions similar to how the Risk Priority Number (RPN) that FMEA produces is used. The calculation to produce EPD for a given flow emission e is provided in Equation 1.

$$EPD_e = P_e * C_e \tag{1}$$

In the EPD equation, the probability P_e, from Table 6, captures all SoS consequence probabilities (either as point probabilities or probability distributions). The cost distribution function C_e, from Table 3, captures all SoS consequence costs (either as point values or cost distribution functions) for a given failure flow emission leaving the system boundary. This is broadly similar to how FMEA produces a RPN to indicate where the largest risks of system failure exist. In the case of EPD, the C_e is similar to the severity of occurrence while the P_e is similar to the combination of the probability of occurrence and detectability of failure.

A generic analysis containing such information is shown in Table 8 where an EPD is produced to help rank order the relative importance of various failure flow system emissions.

Failure Flow System Emission	P_e	C_e	EPD
Energy-Mechanical-Translational	5.2E-4/year	\$5M	\$2600/yr
Material-Solid-Particulate	1.9E-5/year	\$1M	\$19/year
Material-Control-Analog	2.6E-4/year	\$2M	\$520/year
⋮	⋮	⋮	⋮

TABLE 8. EPD Analysis of a generic UAV system that is part of a mixed UAV and UGV SoS. P_e comes from Table 6 and C_e is from Table 3.

STEP 5: IDENTIFY SPURIOUS FLOW EMISSION MITIGATION STRATEGIES

The next step is to develop methods to mitigate failure flow system emissions before they leave the system of interest. In this method, we advocate for addressing such spurious emissions before they leave the system boundary. Other approaches such as hardening other systems within a SoS against unexpected failure flows has been addressed in [23] and elsewhere [92]. We specifically advocate for mitigating failure flow system emissions at the source system in order to for the system to be a "good neighbor" within the SoS. This is in line with guidance in a number of industries and communities of practice such as Federal Communications Commission (FCC) rules on spurious EMF from products [110] and California Air Resource Board (CARB) rules on automotive tailpipe emissions [111]. However, we acknowledge that selfish design leading to "the tragedy of the commons" is a common and unfortunate approach to dealing with undesirable system emissions [112, 113].

Mitigation strategies are likely to be diverse and creative. However, a database of mitigation techniques can be built for a given system where there may be many strategies to mitigate one failure flow system emission or one mitigation strategy that will prevent or reduce the probability of one or more emissions leaving the system boundary. We recommend information on mitigation strategies include both the functional representation and the physical solution to each mitigation strategy. Additionally, information on (1) the likelihood of completely mitigating the failure flow system emission, (2) other failure flows that may be created by the mitigation strategy, and (3) other relevant failure data should be captured at this stage. Table 9 shows a generic example of several mitigation strategies. This is in line with previous work on failure flow arrestor functions [92]. As with Step 1, we advocate that sufficient and significant analysis be undertaken for each proposed mitigation strategy. However, we acknowledge that this may be impractical for practitioners, especially those working at the early phase of design.

In Table 9, P_{Me} is the probability distribution function of a mitigated system failure flow emission still occurring in spite of the mitigation strategy. NFFLS (New Failure Flow Leaving System) represents if a new failure flow may leave the system as a spurious emission. P_{Mf} is the probability distribution function of a new failure flow identified by NFFLS leaving the system. Note that P_{Mf} does not provide insight into if the new failure flow system emission can damage other systems within the SoS. M_C is the mitigation implementation cost distribution function. In all cases where there are either probability or cost distribution functions, we acknowledge that insufficient data early in the system architecture and system engineering process will force practitioners to use point values rather than distributions at least initially. In subsequent iterations of the method as the system design progresses, distributions can be included to gain a more nuanced understanding of the emissions.

In order to understand what mitigation strategies are preferred, we proposed developing a mitigation probability distribution (MPD) which is similar in formulation to EPD. To calculate a population of MPDs where one mitigation strategy may be useful in mitigating several failure flow system emissions (or one emission may be addressed to different extents by different mitigation strategies), a matrix is populated with EPDs that reflect a reduction in P_e, denoted by $\mathbf{EPD_{Reduced}}$. Equation 2 demonstrates how to calculate $\mathbf{EPD_{Reduced}}$ where e is the failure flow system emission and m is the mitigation strategy.

$$\mathbf{EPD_{Reduced}}_{(\mathbf{e,m})} = P_{Me_{(e,m)}} * C_{e_{(e,m)}} \tag{2}$$

An example of the matrix that is populated by Equation 2 is shown in Matrix 1. Here, each row corresponds to a failure emission while each column corresponds to a mitigation strategy. Many of the cells resolve to zero in this matrix, indicating that the mitigation strategy has no impact on that emission.

Next, MPD can be calculated for the matrix produced from Equation 2 as shown in Equation 3. In this equation, $\overrightarrow{EPD}$ represents a column vector of the original EPDs as identified in Table 3.

$$\mathbf{MPD} = \overrightarrow{EPD}^{\intercal} * \mathbf{EPD_{Reduced}} + \mathbf{M_C} \tag{3}$$

$$\mathbf{EPD}_{\mathbf{Reduced_{e,m}}} = \begin{bmatrix} EPD_{Post-Mitigation_{1,1}} & EPD_{Post-Mitigation_{1,2}} & 0 \\ 0 & 0 & EPD_{Post-Mitigation_{2,3}} \\ 0 & EPD_{Post-Mitigation_{3,2}} & 0 \\ EPD_{Post-Mitigation_{4,1}} & 0 & EPD_{Post-Mitigation_{4,3}} \\ \vdots & \vdots & \vdots \end{bmatrix} \tag{1}$$

STEP 6: DETERMINE WHAT MITIGATION STRATEGIES TO IMPLEMENT

In order to determine what mitigation strategies to implement in a system to make it a ”good neighbor” in a SoS, we propose the mitigation rank priority (MRP) metric which converts the probability cost distribution function MDP into a metric than can be more easily rank-ordered. Equation 4 shows how to determine MRP for a specific mitigation strategy, *m*.

$$\begin{aligned} MRP_m = rank(max(MPD_m))+ \\ rank(mean(MPD_m))+ \\ rank(std(MPD_m)) \end{aligned} \tag{4}$$

MRP as presented in Equation 4 is only one potential formulation of MRP, where each term corresponds to the estimated worst case (max), average case (mean), and predictability (standard deviation) of the distribution. Practitioners may wish to change the formulation depending on, for instance, how much confidence they have in their data sources. The important aspect of MRP for the purposes of this method is that it can be used to develop rank orderings and trade space exploration graphics which may be useful to system stakeholders and decision-makers for their understanding of failure flow system emission risks and mitigation strategies. In short: MRP helps in the communication of risk management to stakeholders and decision-makers.

At this point in the method, trade-off studies and optimization can be conducted between major system constraints and requirements, mitigation strategies and their corresponding reduction in probability of a failure flow system emission from leaving the system boundary and adversely impacting other systems within the SoS, and other important system performance metrics. One approach of interest to practitioners is to maximize total MRP (sum of all MRP_m identified for implementation) within the constraint of a cost cap on total mitigation cost (M_C).

In general, the goal of a trade-off study or an optimization is to identify the combination of mitigation strategies from both a functional and a component solution perspective that provides the greatest overall reduction in potential cost to the SoS. While it may be possible to develop a system that does not have the potential to produce any failure flow system emissions that are deleterious to the SoS, the cost of such an undertaking is likely too high to be practical. Instead, a balance must be struck between the cost of mitigation strategies selected, the reduction in cost of the consequences to the SoS, and the available budget to implement the mitigation strategies. It is also important to keep in mind the original failure probability targets set in the Preparatory Step of this method.

STEP 7: ITERATE AND REANALYZE

At this point, mitigation strategies have been chosen and are ready for implementation into the system functional model. We recommend iterating through the method again to verify that the mitigation strategies have not introduced new failures into the system or failure flow system emissions that are undesirable. In particular, Table 9 indicates if there are new failure flows (not quantified in the above method to this point) and failure flow system emissions (P_{Mf}) created by the proposed mitigation strategies. In the preceding steps, these were not addressed in the MRP and earlier calculations.

Re-analysis is further justified by the potential for the failure probability requirements set in the Preparatory Phase being violated from unintended consequences of the mitigation strategies. For instance, a new rotor shroud on a UAV may significantly reduce payload capacity thus violating Requirement #1 from Table 1. The information developed in the first pass through the method can be largely re-used to speed the analysis of the system design by using a software implementation of the method to rapidly develop new iterations of the system design. However, at the end of each iteration, we recommend that a human remain in the loop to verify that results are realistic.

Iteration of the system design through the methodology stops when the specific goals, targets, and requirements set in the Preparatory Step of the methodology are met. At this point, the practitioner can be relatively confident in an exhaustive consideration of potential spurious system emissions having been conducted. Further, a practitioner can be assured that a significant assessment of potential mitigation strategies has been completed. The resulting system design is expected to produce fewer spurious system emissions that may damage other systems within the SoS.

Failure Flow System Emission	Mitigation Strategy Function(s)	Physical Solution(s)	P_{Me}	New Failure Flow(s)	NFFLS?	P_{Mf}	M_C
Energy,Mechanical, Translational	Control Magnitude,Stop,Inhibit	Shielding to prevent rotor strikes	4.7E-5/year	Material,Solid,Object	Yes	3.5E-3/year	\$300k
Signal,Status,Visual	Signal,Process	Redundant control system to verify visual control signals before sending	4.2E-5/year	No	No	0	\$1M
Material,Liquid	Provision,Store, Contain	Catchment subsystem to retain any liquid generated by failed battery cells	5.2E-5/year	No	No	0	\$500k
”	Channel,Export	Long hose to direct liquid to ground	6.2E-5/year	Material,Mixture,Liquid-Solid	Yes	3.1E-5/year	\$250k
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

TABLE 9. Generic mitigation strategies for a UAV. P_{Me} is the probability of a mitigated system failure flow emission still occurring. NFFLS represents if a new failure flow may leave the system from the mitigation strategy function. P_{Mf} is the probability distribution function of a new failure flow leaving the system. M_C is the mitigation cost distribution function.

CASE STUDY RESULT INSIGHTS

The case study demonstrates the types of unexpected insights that this method can uncover and address. The first is that a variety of failure flow emissions may make UAVs in a SoS more detectable by adversaries. The second is that failures in subsystems such as those involved electronic warfare countermeasures may cause failure flow system emissions that have a significant detrimental effect to the other systems in the SoS including a disruption in communications and a higher potential for ”friendly fire” incidents. The method provides a framework to not only identify these emissions and communicate their impacts, but to weigh the tradeoffs between mitigation strategies at the functional stage. This includes the capability compare a portfolio of many cheap mitigation strategies against a portfolio of few expensive mitigation strategies, weigh their relative cost versus effectiveness, and come to a decision.

DISCUSSION

The method presented above has several benefits and drawbacks discussed in this section. Additionally, guidance and discussion on how to implement the method in software is provided.

One significant challenge of the method is the amount of effort required to develop the various database products and analyses. However, we argue that similar efforts are needed for PRA and for other FFIP-based methods. In our experience, PRA analysis can be extremely data-intensive [109].

If this method is more widely adopted, the databases needed to run the analysis can be reused where the data is appropriate to new system and SoS development. A failure and mitigation repository similar to functional design repositories [38] and component failure databases [114] can be developed. Such an effort would also support a computer conducting a greater portion of the analysis and may eventually lead to a fully automated implementation.

One limitation of the method is that it is specifically designed to be used in the case where a practitioner has a good understanding of the SoS that the system being engineered will be placed within. In the case where an entirely new SoS is under development, additional methodological development is needed to manage the uncertainty posed by the situation. If nothing is known of the SoS, C_e cannot be determined. The only information available to a practitioner would then be PO_e.

In order to implement the methodology presented above in software, we suggest using code and methods developed by Short et. al. [96], by Jensen et. al. [89], by O’Halloran et. al. [91], and by O’Halloran et. al. [97] to conduct the FFIP-style failure flow cut-set analysis in Step 2 and the probability quantification in Step 3. The databases developed in the Preparatory Step and in Step 1 can be integrated into the above mentioned code. The equations and tables in subsequent steps can be implemented into the proposed software.

While developing the probability density functions and cost density functions for the method, practitioners may be in a situation where they are very uncertain about the validity of available data. In this case, we have successfully used the Jeffreys Prior [115] in our professional practices and can recommend its use. In essence, the Jeffreys Prior when used in the context of the various probability and cost density functions in this method indicates that there is great uncertainty about the actual distribution.

Validation of the results of the method is an important step that we advocate be performed by a human. We have designed the method to include a human in the loop at every iteration in order to validate that the results are reasonable. While automating the validation may be possible in the future and with a very robust failure and mitigation repository, such an undertaking is very resource-intensive. The method is intended to be used in the system architecture phase of the systems engineering process where rapid trade-off studies and iterations of the proposed design are valued. There are other potential methods of validating the individual elements of the method although such validations have already been published [88, 89, 97, 102, 103]. Validation of the usefulness of alternative functional models is a design process is also in the literature [116]. Thus, we recommend that a human remain in the loop when using the method for the foreseeable future.

The method presented above differs from existing methods of identifying and mitigating spurious system emissions in a SoS context in several ways. Most existing methods such as requirements management, PRA and FMEA, and other similar techniques from the systems engineering community either only implicitly suggest that spurious system emissions be examined and mitigated at the conceptual stage of design before functional architecture has been solidified or explicitly examine spurious systems emissions after functional architectures have been finalized and component design has begun. While an existing functional failure analysis method does look at spurious system emissions, it does so from the perspective of defending against the spurious systems emissions rather than preventing the emissions from occurring. The method introduced in this text specifically takes a quantitative functional approach to analyze spurious system emissions and prevent them from exiting the system boundary to interfere with other members of the SoS. This is different than performing a sensitivity analysis on a functional model or on any existing functional failure analysis method because the method presented above provides insights into what spurious system emissions may occur, how bad the spurious system emissions might be, what to do about the spurious system emissions, and the ranked importance of mitigating spurious system emissions.

CONCLUSION AND FUTURE WORK

This paper presented a conceptual design method intended for use during the system architecture phase of the systems engineering process to identify and mitigate potential spurious system emissions. The method is conducted using functional models which are appropriate for early system architecture trade-off studies. A systematic way to identify potential low probability but high consequence spurious system emissions is presented using the FBED flow taxonomy. Practitioners are aided by the method in identifying and mitigating spurious system emissions to prevent damage to other systems within a SoS.

FUTURE WORK

Several potential fruitful avenues of future work have been identified. A methodology to tie together failure analysis of an SoS that bridges the method presented in this paper and elsewhere [23] may provide a new way of making large system architecture decisions while such decisions are still relatively inexpensive to implement. A potential barrier to such an implementation is the computational efficiency of analyzing many systems using a FFIP-style approach.

Another potential area of future work is to capture preferences of engineers a priori and feed that information into an AI to automatically choose optimal mitigation strategies. Similar work has been conducted by McIntire in a mechanical design context [117]. This is also similar to work done on engineering risk attitudes [118, 119].

A potential area of research to expand on work [92] with failure arresting functions that is useful in the context of the methodology presented in this paper is to examine more complex mitigation strategies and build a mitigation strategy repository. For instance, there may be mitigation strategies that contain spurious emissions in very specific ways so that other systems in the SoS on the receiving end of the spurious emission can robustly handle the failure flow. Another example of a potential mitigation strategy is a mitigation subsystem that can detect an incipient spurious emission and inject recovery flows into failed functions that are about to emit the spurious emissions to prevent the emission in the first place.

An expansion of the analysis presented in this paper could include the idea of flow levels as developed by L'Her et. al. [90]. This may provide a more nuanced view of how far spurious emissions can travel and the amount of damage that they can do within a SoS. Including flow levels may be an intermediate step between the method presented in this paper and a full physics-based simulation of the system and the SoS.

ACKNOWLEDGMENT

This research is partially supported by the Naval Postgraduate School and the Technical University of Denmark. Any opin-

ions or findings of this work are the responsibility of the authors, and do not necessarily reflect the views of the sponsors or collaborators. The case study presented in this publication, while based on real systems, is intentionally fictional and idealized in nature.

REFERENCES

[1] Blanchard, B., and Fabrycky, W., 2006. *Systems Engineering and Analysis*, 4th ed. No. International Series in Industrial and Systems. Prentice-Hall, Upper Saddle River, NJ.

[2] Kapurch, S., 2010. *NASA Systems Engineering Handbook*. DIANE Publishing Company.

[3] Shortell, T. M., 2015. *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. John Wiley & Sons.

[4] Weilkiens, T., Lamm, J., Roth, S., and Walker, M., 2015. *Model-Based System Architecture*. Wiley Series in Systems Engineering and Management. Wiley.

[5] Crawley, E., Cameron, B., and Selva, D., 2015. *System Architecture: Strategy and Product Development for Complex Systems*. Always learning. Pearson.

[6] Van Bossuyt, D. L., Tumer, I. Y., and Wall, S. D., 2013. "A case for trading risk in complex conceptual design trade studies". *Research in Engineering Design,* ***24***(3), pp. 259–275.

[7] Zachman, J. A., 1987. "A framework for information systems architecture". *IBM systems journal,* ***26***(3), pp. 276–292.

[8] Dodaf - dod architecture framework version 2.02. Tech. rep.

[9] Dam, S., 2015. *Dod Architecture Framework 2.0: A Guide to Applying Systems Engineering to Develop Integrated, Executable Architectures*. Createspace Independent Pub.

[10] ISO, 2011. "Iec/ieee systems and software engineering: Architecture description". *ISO/IEC/IEEE 42010: 2011 (E)(Revision of ISO/IEC 42010: 2007 and IEEE Std 1471-2000)*.

[11] Evans, M. K., and Hague, L. R., 1962. "Master plan for information-systems". *Harvard Business Review,* ***40***(1), pp. 92–103.

[12] Kotusev, S., 1986. "The history of enterprise architecture: An evidence-based review". *Journal of Enterprise Architecture–Volume,* ***12***(1), p. 29.

[13] Urbaczewski, L., and Mrdalj, S., 2006. "A comparison of enterprise architecture frameworks". *Issues in Information Systems,* ***7***(2), pp. 18–23.

[14] Ackoff, R. L., 1971. "Towards a system of systems concepts". *Management science,* ***17***(11), pp. 661–671.

[15] Jamshidi, M., et al., 2008. *System of systems engineering: innovations for the twenty-first century*, Vol. 1. Wiley Hoboken.

[16] Boardman, J., and Sauser, B., 2006. "System of systems-the meaning of of". In 2006 IEEE/SMC International Conference on System of Systems Engineering, IEEE, pp. 6–pp.

[17] Sousa-Poza, A., Kovacic, S., and Keating, C., 2008. "System of systems engineering: an emerging multidiscipline". *International Journal of System of Systems Engineering,* ***1***(1-2), pp. 1–17.

[18] DeLaurentis, D. A., Crossley, W. A., and Mane, M., 2011. "Taxonomy to guide systems-of-systems decision-making in air transportation problems". *Journal of Aircraft,* ***48***(3), pp. 760–770.

[19] DeLaurentis, D. A., 2005. "A taxonomy-based perspective for systems of systems design methods". In 2005 IEEE international conference on systems, man and cybernetics, Vol. 1, IEEE, pp. 86–91.

[20] Hause, M. C., 2014. "Sos for sos: A new paradigm for system of systems modeling". In 2014 IEEE Aerospace Conference, IEEE, pp. 1–12.

[21] Dahmann, J., 2015. "The state of systems of systems engineering knowledge sources". In 2015 10th System of Systems Engineering Conference (SoSE), IEEE, pp. 475–479.

[22] Dickerson, C. E., and Jamshidi, M., 2009. "Defense applications of sos". *Systems of Systems Engineering: Principles and Applications*, pp. 319–337.

[23] Van Bossuyt, D. L., OHalloran, B. M., and Arlitt, R. M., 2018. "Irrational system behavior in a system of systems". In 2018 13th Annual Conference on System of Systems Engineering (SoSE), IEEE, pp. 343–349.

[24] Otto, K., and Wood, K., 2001. *Product Design: Techniques in Reverse Engineering and New Product Development*. Prentice Hall.

[25] , 1981. Icam architecture part ii-volume iv - function modeling manual (idef0).

[26] Friedenthal, S., Moore, A., and Steiner, R., 2014. *A practical guide to SysML: the systems modeling language*. Morgan Kaufmann.

[27] Huang, E., Ramamurthy, R., and McGinnis, L. F., 2007. "System and simulation modeling using sysml". In Proceedings of the 39th conference on Winter simulation: 40 years! The best is yet to come, IEEE Press, pp. 796–803.

[28] Fowler, M., 2004. *UML distilled: a brief guide to the standard object modeling language*. Addison-Wesley Professional.

[29] van Eck, D., McAdams, D. A., and Vermaas, P. E., 2007. "Functional decomposition in engineering: a survey". In ASME 2007 International design engineering technical conferences and computers and information in engineering conference, American Society of Mechanical Engi-

neers, pp. 227–236.

[30] Hirtz, J., Stone, R., McAdams, D., Szykman, S., and Wood, K., 2002. “A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts”. *Research in Engineering Design,* ***13***(2), pp. 65–82.

[31] Stone, R. B., and Wood, K. L., 2000. “Development of a functional basis for design”. *Journal of Mechanical design,* ***122***(4), pp. 359–370.

[32] Schmidt, D. C., 2006. “Model-driven engineering”. *COMPUTER-IEEE COMPUTER SOCIETY-,* ***39***(2), p. 25.

[33] Derler, P., Lee, E. A., and Vincentelli, A. S., 2012. “Modeling cyber–physical systems”. *Proceedings of the IEEE,* ***100***(1), pp. 13–28.

[34] Bohm, M. R., Stone, R. B., and Szykman, S., 2003. “Enhancing Virtual Product Representations for Advanced Design Repository Systems”. *Accepted to Journal of Computer Information Science in Engineering*.

[35] Bohm, M. R., Stone, R. B., Simpson, T. W., and Steva, E. D., 2006. “Introduction of a data schema: the inner workings of a design repository”. In ASME 2006 international design engineering technical conferences and computers and information in engineering conference, American Society of Mechanical Engineers, pp. 631–642.

[36] Bohm, M. R., Vucovich, J. P., and Stone, R. B., 2005. “Capturing creativity: Using a design repository to drive concept innovation”. In ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. 331–342.

[37] Bohm, M. R., Vucovich, J. P., and Stone, R. B., 2008. “Using a design repository to drive concept generation”. *Journal of Computing and Information Science in Engineering,* ***8***(1), p. 014502.

[38] Bohm, M. R., and Stone, R. B., 2004. “Product design support: exploring a design repository system”. In ASME 2004 International Mechanical Engineering Congress and Exposition, American Society of Mechanical Engineers, pp. 55–65.

[39] Bohm, M. R., Stone, R. B., Simpson, T. W., and Steva, E. D., 2008. “Introduction of a data schema to support a design repository”. *Computer-Aided Design,* ***40***(7), pp. 801–811.

[40] Van Wie, M., Bryant, C. R., Bohm, M. R., McAdams, D. A., and Stone, R. B., 2005. “A model of function-based representations”. *AI EDAM,* ***19***(2), pp. 89–111.

[41] Stamatis, D. H., 2003. *Failure mode and effect analysis: FMEA from theory to execution*. ASQ Quality Press.

[42] Gilchrist, W., 1993. “Modelling failure modes and effects analysis”. *International Journal of Quality & Reliability Management,* ***10***(5).

[43] Borgovini, R., Pemberton, S., and Rossi, M., 1993. Failure mode, effects and criticality analysis (fmeca). Tech. Rep. AD-A278 508, Reliability Analysis Center.

[44] , 2012. System safety. Standard Practices MIL-STD-882E, Department of Defense.

[45] Segismundo, A., and Augusto Cauchick Miguel, P., 2008. “Failure mode and effects analysis (fmea) in the context of risk management in new product development: A case study in an automotive company”. *International Journal of Quality & Reliability Management,* ***25***(9), pp. 899–912.

[46] Arabian-Hoseynabadi, H., Oraee, H., and Tavner, P., 2010. “Failure modes and effects analysis (fmea) for wind turbines”. *International Journal of Electrical Power & Energy Systems,* ***32***(7), pp. 817–824.

[47] Ericson, C. A., 2005. “Fault tree analysis”. *Hazard analysis techniques for system safety*, pp. 183–221.

[48] Ericson, C., 2005. “Event tree analysis”. *Hazard Analysis Techniques for System Safety*, pp. 223–234.

[49] Stott, J. E., Britton, P., Ring, R. W., Hark, F., and Hatfield, G. S., 2010. “Common cause failure modeling: Aerospace versus nuclear”.

[50] Cornford, S. L., Paulos, T., Meshkat, L., and Feather, M., 2003. “Towards more accurate life cycle risk management through integration of ddp and pra”.

[51] Paté-Cornell, E., and Dillon, R., 2001. “Probabilistic risk analysis for the nasa space shuttle: a brief history and current work”. *Reliability Engineering & System Safety,* ***74***(3), pp. 345–352.

[52] IAEA, 1993. Defining initiating events for purposes of probabilistic safety assessment. Tech. Rep. IAEA-TECDOC-719, International Atomic Energy Agency.

[53] Knochenhauer, M., and Louko, P., 2004. “Guidance for external events analysis”. In Probabilistic Safety Assessment and Management, Springer, pp. 1498–1503.

[54] Siu, N., Mosleh, A., and Meacham, B., 2004. “September 11, columbia, davis-besse, and pra: Business as usual?”. In Probabilistic Safety Assessment and Management, Springer, pp. 1026–1031.

[55] Joksimovic, V., and Vesely, W., 1980. “Use of pra in evaluating safety of nuclear power”. *Reliability Engineering,* ***1***(1), pp. 69–78.

[56] Sorensen, J., Apostolakis, G., Kress, T., and Powers, D., 1999. “On the role of defense in depth in risk-informed regulation”. *Proceedings of PSA,* ***99***, pp. 22–26.

[57] Modarres, M., 2009. “Advanced nuclear power plant regulation using risk-informed and performance-based methods”. *Reliability Engineering & System Safety,* ***94***(2), pp. 211–217.

[58] Stamatelatos, M., Dezfuli, H., Apostolakis, G., Everline, C., Guarro, S., Mathias, D., Mosleh, A., Paulos, T., Riha, D., Smith, C., et al., 2011. “Probabilistic risk assessment procedures guide for nasa managers and practitioners”.

[59] Dezfuli, H., Kelly, D., Smith, C., Vedros, K., and Galyean, W., 2009. "Bayesian inference for nasa probabilistic risk and reliability analysis".

[60] Modarres, M., 2016. *Risk Analysis in Engineering: Techniques, Tools, and Trends*. CRC Press.

[61] Uday, P., and Marais, K., 2015. "Designing resilient systems-of-systems: A survey of metrics, methods, and challenges". *Systems Engineering,* ***18***(5), pp. 491–510.

[62] Madni, A. M., and Jackson, S., 2009. "Towards a conceptual framework for resilience engineering". *IEEE Systems Journal,* ***3***(2), pp. 181–191.

[63] Sheard, S., 2008. "11.2. 2 a framework for system resilience discussions". In INCOSE International Symposium, Vol. 18, Wiley Online Library, pp. 1243–1257.

[64] Carlson, J. M., and Doyle, J., 2000. "Highly optimized tolerance: Robustness and design in complex systems". *Physical review letters,* ***84***(11), p. 2529.

[65] Ross, A. M., Rhodes, D. H., and Hastings, D. E., 2008. "Defining changeability: Reconciling flexibility, adaptability, scalability, modifiability, and robustness for maintaining system lifecycle value". *Systems Engineering,* ***11***(3), pp. 246–262.

[66] Richards, M. G., Hastings, D., Rhodes, D., and Weigel, A., 2007. "Defining survivability for engineering systems". In 5th Conference on Systems Engineering Research, Hoboken, NJ, Citeseer.

[67] Firesmith, D. G., 2003. Common concepts underlying safety security and survivability engineering. Tech. rep., CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.

[68] Henley, E. J., and Kumamoto, H., 1981. *Reliability engineering and risk assessment*, Vol. 568. Prentice-Hall Englewood Cliffs (NJ).

[69] Distefano, S., and Puliafito, A., 2009. "Dependability evaluation with dynamic reliability block diagrams and dynamic fault trees". *IEEE Transactions on Dependable and Secure Computing,* ***6***(1), pp. 4–17.

[70] Birolini, A., 2007. *Reliability Engineering: Theory and Practice*. Engineering online library. Springer Berlin Heidelberg.

[71] Modarres, M., Kaminskiy, M., and Krivtsov, V., 1999. *Reliability Engineering and Risk Analysis: A Practical Guide*. Taylor & Francis.

[72] Tobias, P., and Trindade, D., 1986. *Applied Reliability*. Electrical and Electronics Technology Handbooks. Van Nostrand Reinhold.

[73] Rausand, M., and Høyland, A., 2004. *System Reliability Theory: Models, Statistical Methods, and Applications*. Wiley Series in Probability and Statistics - Applied Probability and Statistics Section. Wiley.

[74] Handbook, E. R. D. Mil-hdbk-338b. us department of defense, 1998.

[75] Düpow, H., and Blount, G., 1997. "A review of reliability prediction". *Aircraft Engineering and Aerospace Technology,* ***69***(4), pp. 356–362.

[76] Wang, W., Loman, J. M., Arno, R. G., Vassiliou, P., Furlong, E. R., and Ogden, D., 2004. "Reliability block diagram simulation techniques applied to the ieee std. 493 standard network". *IEEE Transactions on Industry Applications,* ***40***(3), pp. 887–895.

[77] Rausand, M., 2014. *Reliability of Safety-Critical Systems: Theory and Applications*. Wiley.

[78] Hawkins, P. G., and Woollons, D. J., 1998. "Failure modes and effects analysis of complex engineering systems using functional models". *Artificial intelligence in engineering,* ***12***(4), pp. 375–397.

[79] Tumer, I. Y., and Stone, R. B., 2003. "Mapping function to failure mode during component development". *Research in Engineering Design,* ***14***(1), pp. 25–33.

[80] Stone, R. B., Tumer, I. Y., and Van Wie, M., 2005. "The function-failure design method". *Journal of Mechanical Design,* ***127***(3), pp. 397–407.

[81] Kurtoglu, T., Campbell, M. I., Bryant, C. R., Stone, R. B., McAdams, D. A., et al., 2005. "Deriving a component basis for computational functional synthesis". In ICED 05: 15th International Conference on Engineering Design: Engineering Design and the Global Economy, Engineers Australia, p. 1687.

[82] Roberts, R. A., Stone, R. B., and Tumer, I. Y., 2002. "Deriving function-failure similarity information for failure-free rotorcraft component design". In ASME 2002 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. 121–131.

[83] Tumer, I. Y., and Stone, R. B., 2001. "Analytical method for mapping function to failure during high-risk component development". In Proceedings of the Design Engineering Technical Conferences.

[84] Arunajadai, S. G., Stone, R. B., and Tumer, I. Y., 2002. "A framework for creating a function-based design tool for failure mode identification".

[85] Stock, M. E., Stone, R. B., and Tumer, I. Y., 2003. "Going back in time to improve design: the elemental function-failure design method". In ASME 2003 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. 431–441.

[86] Lough, K. G., Stone, R., and Tumer, I. Y., 2005. "Function based risk assessment: mapping function to likelihood". In ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. 455–467.

[87] Hutcheson, R. S., McAdams, D. A., Stone, R. B., and

Tumer, I. Y., 2007. "Function-based behavioral modeling". In ASME 2007 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. 547–558.

[88] Kurtoglu, T., Tumer, I. Y., and Jensen, D. C., 2010. "A functional failure reasoning methodology for evaluation of conceptual system architectures". *Research in Engineering Design,* ***21***(4), pp. 209–234.

[89] Jensen, D., Tumer, I. Y., and Kurtoglu, T., 2009. "Flow state logic (fsl) for analysis of failure propagation in early design". In International Design Theory and Methodology Conference, IDETC/CIE, ASME, San Diego, CA.

[90] L'Her, G., Van Bossuyt, D. L., and O'Halloran, B. M., 2017. "Prognostic systems representation in a function-based bayesian model during engineering design". *International Journal of Prognostics and Health Management,* ***8***(2), p. 23.

[91] O'Halloran, B. M., Papakonstantinou, N., and Van Bossuyt, D. L., 2015. "Modeling of function failure propagation across uncoupled systems". In Reliability and Maintainability Symposium (RAMS), 2015 Annual, IEEE, pp. 1–6.

[92] Slater, M. R., and Van Bossuyt, D. L., 2015. "Toward a dedicated failure flow arrestor function methodology". In ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. V02AT03A050–V02AT03A050.

[93] Short, A. R., Van Bossuyt, D. L., et al., 2015. "Rerouting failure flows using logic blocks in functional models for improved system robustness: failure flow decision functions". *ICED, Milan, Italy, July*, pp. 27–30.

[94] O'Halloran, B. M., Papakonstantinou, N., and Van Bossuyt, D. L., 2016. "Cable routing modeling in early system design to prevent cable failure propagation events". In 2016 Annual Reliability and Maintainability Symposium (RAMS), IEEE, pp. 1–6.

[95] Short, A.-R., Lai, A. D., and Van Bossuyt, D. L., 2018. "Conceptual design of sacrificial sub-systems: failure flow decision functions". *Research in Engineering Design,* ***29***(1), pp. 23–38.

[96] Short, A.-R., Hodge, R. D., Van Bossuyt, D. L., and DuPont, B., 2018. "Active mission success estimation through functional modeling". *Research in Engineering Design,* ***29***(4), pp. 565–588.

[97] O'Halloran, B. M., Papakonstantinou, N., Giammarco, K., and Van Bossuyt, D. L., 2017. "A graph theory approach to functional failure propagation in early complex cyber-physical systems (ccpss)". In INCOSE International Symposium, Vol. 27, Wiley Online Library, pp. 1734–1748.

[98] Dempere, J., Papakonstantinou, N., O'Halloran, B. M., and Van Bossuyt, D. L., 2018. "Risk modeling of variable probability external initiating events in a functional modeling paradigm". *The Journal of Reliability, Maintainability, and Supportability in Systems Engineering,* ***Summer 2018***, pp. 5–16.

[99] Van Bossuyt, D. L., O'Halloran, B. M., and Papakonstantinou, N., 2019. "A system design method to reduce cable failure propagation probability in cable bundles". *The Journal of Reliability, Maintainability, and Supportability in Systems Engineering,* ***Winter 2018-2019***, pp. 5–12.

[100] OHalloran, B. M., Papakonstantinou, N., and Van Bossuyt, D. L., 2018. "Assessing the consequence of cyber and physical malicious attacks in complex, cyber-physical systems during early system design". In 2018 IEEE 16th International Conference on Industrial Informatics (INDIN), IEEE, pp. 733–740.

[101] Coldren, L. O., 1985. "Afghanistan in 1984: the fifth year of the russo-afghan war". *Asian Survey,* ***25***(2), pp. 169–179.

[102] Arlitt, R., Van Bossuyt, D. L., Stone, R. B., and Tumer, I. Y., 2017. "The function-based design for sustainability method". *Journal of Mechanical Design,* ***139***(4), p. 041102.

[103] Arlitt, R., and Van Bossuyt, D. L., 2019. "A generative human-in-the-loop approach for conceptual design exploration using flow failure frequency in functional models". *Journal of Computing and Information Science in Engineering*.

[104] 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), pp. 051401–051401.

[105] O'Halloran, B. M., 2013. "A framework to model reliability and failures in complex systems during the early engineering design process".

[106] Ullman, D. G., 2017. *The Mechanical Design Process*, 6th edition ed. David Ullman LLC.

[107] Taleb, N. N., 2007. *The Black Swan: The Impact of the Highly Improbable*. Random House.

[108] Taleb, N. N., 2007. "Black swans and the domains of statistics". *The American Statistician,* ***61***(3), pp. 198–200.

[109] Westinghouse Electric Company, LLC, 2004. *AP1000 Probabilistic Risk Assessment*, revision 7 ed. Pittsburgh, PA.

[110] Vikstedt, J., 2010. Radiated spurious emission testing.

[111] Yang, C., McCollum, D., McCarthy, R., and Leighty, W., 2009. "Meeting an 80% reduction in greenhouse gas emissions from transportation by 2050: A case study in california". *Transportation Research Part D: Transport and Environment,* ***14***(3), pp. 147–156.

[112] Hardin, G., 1968. “The tragedy of the commons”. *science,* ***162***(3859), pp. 1243–1248.

[113] Feeny, D., Berkes, F., McCay, B. J., and Acheson, J. M., 1990. “The tragedy of the commons: twenty-two years later”. *Human ecology,* ***18***(1), pp. 1–19.

[114] OAK RIDGE NATIONAL LABORATORY, 1982. *The In-Plant Reliability DataBase for Nuclear Power Plant Components: Data Collection and Methodology Report*, nureg/cr-2641 ed. Washington, DC.

[115] Jeffreys, H., 1998. *The Theory of Probability*. Oxford Classic Texts in the Physical Sciences. OUP Oxford.

[116] Shah, J. J., Smith, S. M., and Vargas-Hernandez, N., 2003. “Metrics for measuring ideation effectiveness”. *Design studies,* ***24***(2), pp. 111–134.

[117] McIntire, M. G., 2016. “From functional modeling to optimization: Risk and safety in the design process for large-scale systems”. PhD thesis, Oregon State University.

[118] Van Bossuyt, D., Hoyle, C., Tumer, I. Y., and Dong, A., 2012. “Risk attitudes in risk-based design: Considering risk attitude using utility theory in risk-based design”. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* ***26***(4), p. 393406.

[119] Van Bossuyt, D. L., Dong, A., Tumer, I. Y., and Carvalho, L., 2013. “On measuring engineering risk attitudes”. *Journal of Mechanical Design,* ***135***(12), p. 121001.