RISK ANALYSIS FRAMEWORK FOR UNMANNED SYSTEMS

A Dissertation
Presented to
The Academic Faculty

By

Joel Dunham

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology

August 2020

RISK ANALYSIS FRAMEWORK FOR UNMANNED SYSTEMS

Approved by:

Dr. Eric Johnson, Advisor
School of Aerospace Engineering
Pennsylvania State University, Georgia Institute of Technology

Dr. Eric Feron, Advisor
School of Aerospace Engineering
Georgia Institute of Technology, King Abdullah University of Science and Technology

Dr. Brian German, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Amy Pritchett
School of Aerospace Engineering
Pennsylvania State University

Dr. John-Paul Clarke
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Maxime Gariel
Chief Technology Officer
Xwing

Date Approved: May 15, 2020

ACKNOWLEDGEMENTS

I would like to thank Dr. Eric Johnson for his role as my primary advisor and for access to his GUST (Georgia Tech UAV Simulation Tool) flight management software, which has been instrumental in testing this research. I further would like to thank Dr. Eric Feron for his role in co-advising me, as his guidance in this field and current research has been invaluable. I would also like to thank Dr. Brian German who has helped to co-advise me as a local advisor while my other advisors have found their places at other universities. I would like to thank Olivia Jagiella-Lodise who provided suggested edits for this dissertation. Finally, I would also like to thank Roy and Julianna Burke who helped facilitate and record the discussions in the 2018 UAS Safety Symposium.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

Belief Mass Non-negative measure similar to probabilities, but a non-classical idea in which the masses are not necessarily based on the occurrence of an event.

Frame of Discernment (Θ) A set of mutually exclusive elements. Belief masses can be assigned to these elements or to sets of these elements.

Powerset (2^{Θ}) The set of all subsets of Θ. If there are n elements in Θ, then there are 2^n elements in the powerset of Θ.

Basic Probability Assignment (BPA) Assignment of a belief mass in the range $[0, 1]$ to each element of the powerset. $m : 2^{\Theta} \rightarrow [0, 1]$, where $m(\emptyset) = 0$, $\sum m(A) \geq 0$, $A \in powerset$. If $\sum m(A) = 1$ and the masses are based on the occurrence of an event, then these masses are equivalent to probabilities. Subsequently, we will assume $\sum m(A) = 1$.

Evidence An observation described by a BPA.

Focal Point Any subset of the powerset to which a belief mass of greater than zero is assigned. $A \in powerset \mid m(A) > 0$

Belief Function (Bel) Given a BPA with $\{A_1, \ldots, A_n\} \in 2^{\Theta}$ and $A_x \in \Theta$, $Bel(A_x) = \sum_{A_x \subseteq A} m(A_i)$.

Plausibility Function (Pl) Given a BPA with $\{m(A_1), \ldots, m(A_n)\} \in 2^{\Theta}$ and $A_x \in \Theta$,

$Pl\left(A_x\right) = 1 - \sum_{A \cap A_x = \emptyset} m\left(A_i\right).$

Dempster's Rule The original rule proposed by Arthur Dempster [1] for combining evidence $\left(m_1 \oplus m_2\right)(x) = \frac{\sum_{E \cap E' = x} m_1(E) m_2\left(E'\right)}{1 - \sum_{E \cap E' = \emptyset} m_1(E) m_2\left(E'\right)}$.

SUMMARY

Airspace regulatory agencies, such as the Federal Aviation Administration (FAA) for the United States (US), are currently focusing on risk assessment frameworks for integrating the operation of Unmanned Aerial Systems (UAS) into National Air Space (NAS). Multiple frameworks, such as the Specific Operations Risk Assessment (SORA) [2] framework for the European Union and similar frameworks for the US, provide defined pathways to evaluate the risk and seek approval for UAS operations. These frameworks are primarily qualitative and are sufficiently flexible to incorporate quantitative approaches, many of which have been proposed and tested in literature. Most proposed quantitative methods are still under development. Likewise, real-time analysis methods, designed to provide decision-making to unmanned systems during operations, have been proposed. Current real-time analysis methods still suffer from limitations, such as only applying to specific operations. This research applies Dempster-Shafer theory and valuation networks [1] [3] [4], a framework for reasoning with uncertainty used extensively for risk analysis, to UAS risk analysis by creating extensions which allow this framework to learn risk relationships in the UAS ecosystem based on operational results and enable this framework to be used in real-time analysis onboard small UAS. These extensions are applied to an autonomous car scenario for testing the capabilities against known baselines, then applied to the UAS scenario for testing in simulation against a previously implemented real-time health monitoring system. Finally, these extensions are demonstrated in flight on a small UAS. Application to the UAS ecosystem and conclusions are addressed based on the results of these tests.

CHAPTER 1

INTRODUCTION

Unmanned Aerial Systems (UASs) are continuing to proliferate rapidly [5]. Engineering development is focused on professional products for cargo delivery, news coverage, mapping, agriculture, and rescue missions, among other uses. Event38 [6] sells commercial-grade systems for agriculture use, which provide a variety of analyses to farmers, previously only available at considerably higher expense and much slower update rates. Consumer products are becoming pushbutton systems which provide a capability to operators without the operators requiring understanding of how the system works or what could go wrong. The DJI Mavic Pro [7] represents a near-entry-level consumer product (approximately $900) with long flight times (approximately 28min), pushbutton operation, automatic handling of common issues such as near seamless switching between GPS navigation and optical flow navigation in the event that the GPS fails, smooth take-offs and landings that do not require user knowledge of vehicle limits, and geo-fencing with knowledge of Temporary Flight Restrictions (TFRs) [7], drastically reducing the requirements for operator knowledge of both current flight conditions and regulations. With this reduction of required operator knowledge, it becomes even more necessary for the system to appropriately handle contingencies without input from the operator since many operators will no longer have the required knowledge to handle contingencies. Furthermore, by removing the operators from direct control over the system, even operators with sufficient knowledge to handle contingencies may not have the required control to be able to do so, thus necessitating the system to handle the contingencies automatically or with some guidance from the operator.

Current efforts are focused on aligning provable safety systems with related regulations to integrate unmanned systems with manned systems in the National Air Space (NAS), allowing them to operate in the same airspaces. In recent years — recent months for many of these advancements — great strides have been made both on the regulatory side and also on the safety technology side. Several risk analysis frameworks have been proposed that have backing from one or more regulatory bodies. Joint Authorities for Rulemaking of Unmanned Systems (JARUS) guidelines on Specific Operations Risk Assessment (SORA) [2] develops a framework in which Unmanned Aerial Systems (UAS) can fly specific operations in a variety of airspaces once preliminary risk assessment and, if necessary, risk mitigations have occurred and been approved by the governing authority. This framework is flexible and enables a UAS operator to tailor the approach of risk-based operation approval to almost any situation. However, while flexible, many gaps still remain, precluding access to many specific operations since the risk reduction requirements are too demanding for current technologies or methods of risk reduction to meet. The Federal Aviation Administration (FAA) Safety Management System (SMS) [8] is another such framework designed to provide a top-down organization-wide management of safety risk and assuring the effectiveness of safety controls. Beyond UAS, this framework applies to all aviation and obliges organizations to manage safety at the same level as all other aspects of the core business processes. A second FAA initiative, Compliance Program [9], is based on the assumption that many errors are honest mistakes and should be self-reported. Mistakes that are self-reported are corrected via a problem-solving approach rather than punished. This philosophy emphasizes a "just culture" [9]. Like SMS, this risk-based approach is focused on the culture that extends through both corporations and the entire aviation industry, but it does not provide a framework for assessing risk of a particular operation or flight plan. Effectively, these are all-encompassing frameworks which do not provide the low-level details necessary to make a risk assessment for particular operations. Together, these frameworks and philosophies form a risk management-based approach to aviation

safety which could provide the regulatory backing to enable UAS integration into NAS.

Technological solutions to safety issues must be consistent with the regulations used to enforce them. Previous methodologies focused on specific technology requirements to access airspace. For example, sense-and-avoid was seen as a must-have capability for UASs intent on flying in the NAS [10]. After 20+ years of research with no accepted solution, the focus has shifted to analyzing the risks associated with the integration of various technologies and determining whether the risk is sufficiently low to perform the given operation or flight plan, such as in the case of SORA [2] or the current FAA focus [11]. The challenge partially shifts to the technologies providing the necessary data on whether they are reliable. For example, fault detection systems are a large field in aerospace research. Concepts such as neural networks for detecting faults in IMUs [12] enable fault detection and handling for a sensor of which failure can cause complete loss of control on a multi-rotor. Sufficient tests of this system result in statistics for the reliability of detection and handling, which provide the necessary data for quantitative risk assessment. Likewise, research on emergency path planning for UAS also exists [13]. These designs provide solutions to the problem of how to respond once the emergency has been identified.

Recent research has enabled offboard real-time risk analysis for UAS, based on current data, Bayes belief networks, and most-likely-hypotheses [14]. Further, consistent separation analysis using avoidance volumes for aircraft based on the precision of navigation systems [15] provide methods of analytically predicting problems and dealing with them as they arise. These advances serve to underscore the importance of and focus on technological solutions to safety systems on the UAS that are consistent with the regulations that govern use of the UAS. Limitations in each of these methods provide part of the basis for the research in this proposal. Specifically, a complete onboard real-time system sufficient for small UAS is currently not operational. Further, quantification of risk through

a Dempster-Shafer analysis [1] [3] provides decision criteria beyond the criteria provided by a Bayesian analysis, which better align with risk-based decisions. A scenario has been identified that covers a broad range of possible system failures and degradations ranging from vehicle issues, such as control or navigation system degradations, to sensor degradations due to external effects. The goal of this research is to develop a risk assessment model which connects risk evidence through long-term operation results analysis with real-time predictive actions to mitigate unacceptable risk during UAS operations. A further requirement of this research is that the chosen risk assessment methodology must be flexible to incorporate constantly changing data about UAS risks while the results must also be explainable to governing authorities (i.e. there is a clear relationship between risk factors and resulting risk assessment that a decision-maker can understand and follow), enabling this analysis to be a potential basis for operational authorization.

1.1 Summary of Contributions

The contributions of this work are as follows:

- Organized and ran the Safety Symposium for Unmanned Aerial Systems in August 2018, bringing together experts from UAS law, regulations, insurance, operations, and research and development to understanding issues with integrating UAS into national airspace, motivating some of this work.
- First proposed use of evidence inputs at nodes to update conditional mass distributions in a Dempster-Shafer network, a model developed by Shenoy [16], Shafer [17], and Smets [4].
- Developed a new algorithm consistent with the above that updates Dempster-Shafer network conditional mass distributions based on observations at nodes in an optimal manner, reducing expert information requirements for these networks to function.

- Defined a new weighting scheme for entering multiple evidence observations simultaneously into a Dempster-Shafer network and updated current Dempster-Shafer combination algorithms [18] [19] [20] to use the weighting scheme, resulting in correctly recording the "experience" captured in a Dempster-Shafer network.
- Created episodic learning for Dempster-Shafer networks, using observability concepts to improve the optimization algorithm discussed above, resulting in more accurately capturing the relationships in the network.
- Applied novel Dempster-Shafer network updates to a simplified autonomous car decision, demonstrating improvements over baselines including researched open-loop profiles [21] and Bayesian Belief Network [22] decisions between these open-loop profiles.
- Applied novel Dempster-Shafer network updates to UAS real-time risk analysis based on Safety Symposium outcomes, proving ability to reduce risk on small UAS through in-flight decisions in simulation. Demonstrated clear improvement over baseline system [23]. Implemented on real-time flight hardware, demonstrating through flight test the ability to provide in-flight risk analysis and decisions on critical fight hardware in small UAS.
- Developed UAS ecosystem model, demonstrating a method through which flight operation experience could be shared for UAS risk analysis.

1.2 Organization of Thesis

This document is organized as follows. Chapter 2 describes the recent state of UAS operations, regulations, and insurance based on expert information obtained through a UAS safety symposium hosted in 2018, along with corroborating literature. This chapter sets the context for the theory development and application described in the subsequent chapters.

Chapter 3 extends the proposed theory to be used as the basis for the common infrastructure for risk assessment of UAS operations. Chapter 4 applies the proposed to an autonomous driving vehicle situation — evaluating a traffic light to determine whether to slow down when approaching a red light. This application tests the theory extensions in a known, understandable scenario to evaluate the ability of the theory extensions to analyze complex situations in a real-time context. Chapter 5 applies the theory extensions to UAS — a news multirotor scenario evaluating the risk associated with remaining on station. Chapter 6 concludes the discussion and provides trajectories of future research to be based on the outcome of this research. Finally, the included appendices A through F include additional discussions, proofs, and notes that are not central to the work, but are necessary for a full understanding.

CHAPTER 2

BACKGROUND

In August 2018, a UAS safety symposium was held in Atlanta, GA, which included experts from many facets of UAS operations in the United States including law, insurance, operations, research, and regulation. While some changes to the UAS landscape have changed since that symposium, much of the principles remain the same. This chapter details those principles, highlights some of the changes since then, and corroborates some of those expert opinions with additional literature, setting the stage for research which could fit into this ecosystem. Note that this symposium was primarily focused on intended legal operation of UAS; while some information and artifacts were discussed with regards to deliberate, malicious, illegal activities, representatives agreed that a different set of principles were required to handle these activities, and those principles were outside the domain expertise of the representatives in attendance. Appendix F includes the raw notes from the safety symposium. Sections 2.1, 2.2, 2.3, 2.4, and 2.5 are primarily written based on expert opinions from the safety symposium as documented in Appendix F. As such, those notes are not constantly referenced throughout the following sections. Corroborating research is cited in these sections.

2.1 Summary of Safety Symposium

By far, the principle take-away was the lack of information concerning use of UAS legally and within regulations. Interestingly, this area is one in which there was significant divergence between regulators and the rest of the industry. Operator, insurance, and law representatives agreed that there was lack of information concerning UAS regulations and

laws, yet the regulatory agency representative provided multiple means of obtaining information about regulations including through Part 107 [24]. Since then, the two ends of the spectrum have grown closer, particularly as the FAA closes towards regulations that will require UAS to identify themselves and transmit location information while operating [11] [25]. Second to this take-away was the consensus that the industry is in wait-and-see mode, especially on the law and regulations side, until precedents are set, usually through a major incident. Again, regulations have been pushing forward as mandates have been created for the FAA to develop regulations for UAS operations [26]. All representatives agreed that enforcing regulations is difficult at best until a reliable means of identifying UAS is available. As mentioned previously, FAA regulations are going into force that will require operating UAS to transmit ID and location information [11]. Finally, there was agreement that operator education is currently lacking. Improvements have been made for Part 107 operation education [24], but there is limited education beyond that point.

2.2 United States Separation of Operations

Unmanned systems currently fly under accommodation practices — relying on operational segregation to avoid issues with manned traffic [27]. As such, they are restricted in use by 14 CFR Part 107, which is applicable for commercial use of small UAS under 55 lbs [24]. Note that UAS regulation is a purely federal affair, since the federal government regulations airspace from the tips of the blades of grass exposed to the outside up as high as the airspace extends. This jurisdiction was solidified as of Boggs versus Meredith in 2015. Part 107 regulation restricts flight to less than 400 ft AGL unless within 400 ft of a higher structure. Operations up to class B airspace [28] are permitted via Part 107 with Air Traffic Control (ATC) permission or in class G airspace with no permission required. All operations under Part 107 must be within line-of-sight of the certified operator who must retain visual sense-and-avoid capabilities over the aircraft, further limiting the available

flight times to daylight or civil twilight and to one pilot per unmanned system. Aircraft flying under Part 107 are not allowed to fly over people not directly involved in the flight of the vehicle. These derived restrictions all come from one basic concept: operational segregation with human decision making as the immediate and final authority for all UAS. These regulations were finalized as of August 29, 2016. While these regulations allow small UAS to operate in national airspace in principle, the practical result is far from the fully integrated vision of researchers, manufacturers, and the FAA alike. Subsequent to this set of regulations, updates have been made, in large part spurred by UAS operations and planned operations that have forced regulations to move forward [26]. Likewise, efforts are being made in partnerships with the FAA and various UAS community stakeholders to develop UAS Traffic Management (UTM) National Campaign II, enabling access to low-altitude airspace for UAS [29]. Various efforts have been made to develop requirements for this airspace integration [30].

For larger UAS — above 55lbs — the only available regulation under which to fly, outside of FAA-designated test sites, is Section 333 [31]. Section 333 deals with waivers/exemptions, otherwise known as Certificate of Waiver or Authorization (COA). This regulation more clearly exemplifies the operational segregation by stating that any unmanned system can be flown in the area under the COA with advanced notice to ATC.

While allowing unmanned research to continue with the goal to develop fully integrated systems, these regulations clearly keep unmanned systems separate from fully integrated national airspace. Due to this segregation, however, unmanned systems do not require any type of certification. Only pre-flight checks by the operator are required under Part 107 [24]. Because many small UAS are never intended for fully integrated flight in national airspace, these regulations enable low cost entry into the small UAS market, without the need for costly certification processes or pilot training. Conversely, all UAS are limited

to these operational restrictions due to the inability to certify systems for flight in fully integrated airspace, precluding manufacturers and operators from developing and utilizing UAS for many operations.

2.3 Laws Applicable to UAS Operations

The United States operates under common law [32]. This means that the body of law arises from precedents having been derived from judicial decisions of courts and/or similar entities [32]. As such, US laws always run behind technology, waiting for a situation in which they will be interpreted to set the precedent for how the laws apply to that technology. Note that laws do not chase the technology; they chase the underlying issues which apply to the technology. Those issues and laws are interpreted with respect to the specific technology. As mentioned previously, Boggs vs. Meredith in 2015 set the precedent for establishing that the FAA has jurisdiction down to the blades of grass if exposed to the outside. Beyond that legal precedent, much of the law concerning UAS falls under nuisance laws such as Peeping Tom laws. For example, video requires consent from both parties — the party recording the video and the party being videoed. Since most UAS carry onboard cameras, UAS operators have to be careful of the field of view of the camera since video footage captured by those cameras are subject to two-party consent.

Since major incidents with UAS have not yet set a precedent, discussions beyond the precedents described above are based on hypothetical situations. Given a hypothetical situation in which a UAS causes the crash of a passenger jet by destroying engines, causing loss of life, the general consensus of the experts at the safety symposium was that everyone gets sued in a civil case — the airlines, the manufacturers, the UAS operator, etc. The burden of proximate cause is placed on the courts to figure out. In reality, this situation likely means that the defense is tendered to the insurance companies, who likely settle. The effect of the

insurance will be discussed in Section 2.4. The most difficult part of this process will be finding enough parts of the UAS to identify the operator. Required UAS ID and tracking — currently in the process of becoming regulations — will significantly simplify this process. However, this does assume all systems are functional, the operator is attempting to fly legally, and position updates between UAS in near vicinity of each other are sufficiently unambiguous along with recovered parts to conclusively point to the system at fault. In many cases, such as UAS crashing when flying near crowds or UAS operating in the way of emergency systems, unsafe operations are often traced through social media postings. Even when proof is found, proper chain of custody procedures must be followed. Often, UAS violations are called in to local law enforcement as first responders. However, since local law enforcement does not have jurisdiction over the airspace, this path is a dead end, usually resulting in the UAS and operator being long gone before law enforcement officials arrive. In summary, likely most UAS violations — other than clear criminal cases — will result in lawsuits that must be allocated by the court system and handled through insurance, assuming that sufficient evidence is available to track down the at-fault parties and the correct authorities are involved.

As can be clearly seen in the preceding paragraphs, there is still much ambiguity in the realm of UAS law and the effects of lawsuits. As such, many operators, especially commercial operators, have resorted to obtaining insurance to handle cases in which lawsuits are the primary recourse by offended parties.

2.4 UAS Operation Insurance

UAS insurance is a legal issue since insurability brings in a host of concerns including violation of FAA rules, physical damage and bodily injury, nuisance laws, trespass laws, invasion of privacy, stalking and harassment, and wiretap laws. Of those, trespass laws

and invasion of privacy laws are two of the major reasons for UAS violations. While there are multiple insurance policies that can be used to cover UAS operations, that is beyond the scope of this paper. Refer to [33] for more details concerning methods to insure UAS. Rather, of interest is high level breakdowns in methods of insurance and how they map to UAS operations. Insurance tends to be evaluated in one of two ways:

- It falls into a "normal" bucket. In this case, it is often passed along to re-insurers who insure/price it based on standard rates. Typically, this is fully automated/computerized with human oversight.
- Some parameters of the insurance request are outside the norm. In this case, it is typically evaluated by a human in the primary insurance companies who helps to determine the risk model and pricing.

Re-insurance, such as Swiss Re, accounts for 65% of recoverables from non-US companies. While often less well-known than primary insurers, re-insurance is a method of spreading risk — a way for primary insurers to insure policies with well-understood risks. To create these buckets of insurance policies that can be passed on to re-insurers, some insurers use exclusions to keep operators within the bounds they specify. However, this method often leads to unintended consequences. Global Aerospace — a primary UAS insurer — removed exclusions because crashes invariably break at least one exclusion regardless of the operation, making the insurance useless if the exclusions were in place. Without exclusions, pricing insurance premiums becomes much more dependent on information from the operators. In fact, the current insurance model is supposed to be on a per-flight basis, but it is often not executed this way.

In summary, the insurance industry already has a model of providing insurance which applies to, or at least overlaps with, UAS. The more pressing issue is that UAS risks are not well understood yet, resulting in standard models still adapting to the UAS model. Fur-

ther, aligning the risk models used by UAS operators and the FAA with the insurance risk models will make the UAS ecosystem integration easier.

2.5 UAS Operation Safety

Operational risk and, conversely, safety closely tie to the question of insurance. In this section, the outcomes of the 2018 safety symposium will be discussed. Much research has also been performed for technology in this area, which will be discussed in Section 2.6. Firstly, there are two classifications of safety that are of interest here: actual safety and perceived safety. Currently, the concept of actual safety appears to be safety from physical harm, applying primarily to humans. Perceived safety has been shown to vary based on velocity, size, and distance of the UAS, with the primary determinant being velocity. The research into perceived safety has been done by interdisciplinary teams using measures of skin conductivity, head tilt, and heart rate, which seems to equate perceived safety with a physical fight or flight response. Perceived safety violations are the primary driver of reports to law enforcement and complaints to regulators versus actual safety violations. Perceived safety is affected more through education and marketing, such as the team referenced in Appendix F that used assistance from a product design and art team to make UAS look more “friendly”.

This research is not focused on changes to perceived safety that are effected through visual product design. Rather, this research focuses on technology changes that can be made to enable improvements in both actual and perceived safety through risk analysis of the UAS ecosystem. The safety symposium experts suggested that additional divisions of weight classes for UAS could offer more refined risk categories for the ecosystem. Categories based on weight and/or max speed could be appropriate since weight and speed directly correlate to kinetic energy, which has a significant impact on whether the UAS can harm

humans or property [14] [34] [35]. While these categories may have less effect on a numerical risk analysis, the experts also suggested additional segmentation for operators based on vehicle classes, beyond Part 107 [24] training, which could affect numerical analysis more strongly since operator capability to handle an emergency could be taken into account. The question raised here is whether different UAS (size and type) have vastly different flight and operational characteristics, or whether the mission and flight control software removes much of the differences in dynamics from the operator.

Regardless of how the risk and safety are determined, the panel believed that commercial use will drive regulations. Losses will come first, which will drive public opinion, in turn driving new regulations. In the meantime, the industry is in a wait-and-see mode. Additionally, there are many redundancy and reliability issues that must be addressed for vehicles and software. Regulation at the manufacturer level may be required to handle safety issues. Unfortunately, software reliability drives cost. Thus, increased reliability requirements may drive costs too high for many UAS manufacturers. An alternate view is that product liability will drive increases in safety and decreases in risk. Consider, for example, operations only being authorized if the risk (determined for each operation) is sufficiently low. Further, consider that even if the operation is authorized, the operators must transfer the risk (through insurance) to adequately protect themselves. Now, the risk is determined both by the authorizing agency and the insurance company. Too high of risk means no authorization to operate. Likewise, even with authorization, too high of risk means the risk transference cost is too high. Consequently, operators will require more reliable systems or alternate methods of operation which lower the risk, thereby driving a reduction in risk without regulation at the manufacturer level. By driving risk down through this method, manufacturers who are unable to meet software reliability requirements for lower risk may be able to decrease risk via other means, perhaps through operational requirements for greater separation from humans and property. Likely, a combination of the above methods

will lead to sufficiently reduced risk to enable operations.

2.5.1 Quantitative Framework Characteristics

Based on the results from the safety symposium, three primary characteristics were derived for a quantitative risk-analysis framework for UAS decision-making:

- Explainable: the risk analysis framework will be used by both regulatory agencies for authorizing operations and by insurers for insuring operations. As such, any quantitative results from a risk analysis framework must be explainable to humans overseeing the operational decisions. This characteristic rules out purely data-driven methods such as neural networks [36] which are flexible but difficult to explain.

- Flexible: as noted in Section 2.1, there is significant unknown with regards to UAS. Coupled with the fact that new UAS are being manufactured and introduced to the ecosystem on a regular basis, the risk analysis framework must be flexible to handle an ever-changing analysis. This characteristic rules out more cognitive systems such as expert systems and case-based reasoning [37] that are easy to explain but inflexible.

- Initialize from unknown: Unlike manned aviation, UAS introduced to the ecosystem are typically flown before risks are quantitatively established — i.e. there are no priors for initializing quantitative frameworks. As such, a framework that can incorporate these UAS quickly must handle lack of *a-priori* information.

2.6 Current Risk Frameworks and Technology

For some specific operations, such as ones that are already authorized under current UAS rules, some current UAS technologies are sufficiently provable for authorization under proposed risk assessment frameworks. Others, however, do not provide the assurances re-

quired to operate in more complex and higher risk environments. In manned aviation, there is always the fall-back position to the human pilot and the airframe. The airframe, which includes critical vehicle systems such as required flight controls, is proven to reliability standards. If all non-critical functions such as sensor failure, the human pilot is believed to be able to recognize the failure and replace the failed system with their own capabilities, thereby providing the ultimate backup system. In helicopters, human-controlled auto-rotation is available. Manned quadrotors are not typically flown, and multirotors with more rotors (such as 18 rotor systems which have been recently demonstrated [38]) can lose at least one motor and still be brought safely to the ground.

For unmanned systems, the technology systems become more crucial since a ground-based pilot might rely on sensors to provide critical flight feedback, the system may rely on decision-making systems to make appropriate recommendations or decision, or, in many cases, the system may rely on flight control systems to control the vehicle even if the operator takes as direct control of the vehicle as possible. For most small UAS, the development of these flight control and sensor systems are not regulated or verified nearly as rigorously as for manned aviation. The question then arises: will UAS be cost effective while proving that they will remain within safety constraints? For some operations, the answer is "yes" since the revenue of those operations outweighs the development and operating costs. For other operations, the answer is likely "no".

A second approach taken by several innovative systems such as Xavion [39] is to create a backup system that does not depend on the primary system. Xavion [39] does so by providing a flight trajectory guidance that the pilot can follow to an alternate landing. Systems that takes over in the event of failure of the primary systems can be developed under guidance from ASTM F3269 [40] (such as an auto ground collision avoidance system). In this case, a single, well-developed backup system could be cost effective by spreading

the cost across the myriads of UAS which would use it. However, this method too has its drawbacks. Either the system must be sufficiently self-contained such that all sensors, controls, actuators, etc. are highly proven and are carried as a second system onboard the vehicle (which is untenable for small UAS), or certain parts of the original system must be trusted. Herein lies the deeper issue: either the original system must accurately report when it is failing, or the backup system must have sufficient technology to detect and handle the failure.

Current research provides means to detect and handle certain failures. In particular, fault detection systems [12] [41] provide detection of failures with some degree of reliability. Further, risk assessment systems [14] provide means to predict the results of failing subsystems and provide the operator with an assessment of the implications. In both cases, some level of *a-priori* knowledge is required. The reliability of the fault detection systems must be characterized in order to feed the priors of the Bayes reasoning-based risk assessment systems [14]. Therein lies the issue: there is insufficient flight time and testing performed on these systems to provide accurate priors, leading groups such as JARUS to fall back to qualitative assessment methodologies in SORA [2]. Subsequent sections discuss research into risk analysis frameworks and underlying technology used to feed the data required for the risk analysis frameworks to function.

2.6.1 Risk Framework Research

Much research has already been performed on risk frameworks, with various papers focusing on individual operations to overall frameworks designed to incorporate technical risks analysis in various aspects of UAS operations. Several papers based on research performed by Roland E. Weibel are focused on risk analysis and mitigated methods for integrating UAS into the US National Airspace System (NAS). Three papers in particular are “An Integrated Approach to Evaluating Risk Mitigation Measures for UAV Operational Con-

cepts in the NAS" in 2005 [42], "Safety Considerations for Operation of Unmanned Aerial Vehicles in the National Airspace System" in 2005 [43], and "Safety Considerations for Operation of Different Classes of Unmanned Aerial Vehicles in the National Airspace System", his Master's Thesis in 2005 [44], all co-authored by R. John Hansman, Jr at MIT. While each of these papers deal with the topic of risk assessment and mitigation for UAS, these papers primarily form what could be seen as the basis of the SORA concept since these papers preceded SORA. These papers do incorporate a different focus in that they are interested in US NAS while SORA is focused on the European airspace [2]. However, their primary concern is still a holistic risk model. As such, grounding risk analysis research in this type of framework applies to US and European airspaces, making this a solid foundation for research.

More recent work develops risk assessment tools designed to be trusted as the basis for FAA assessment and operator evaluation [45]. This work focuses on the numerical risk assessment of air and ground collisions, using historical data from manned aircraft to validate the model. Further, this work relates the risk to costs to insure the operation in order to make the results more meaningful. Similarly, recent work in characterizing the consequences of UAS collisions in the NAS [46] provide an improved understanding of the risks, paving the way for integration into a comprehensive risk analysis framework. Other recent risk assessment frameworks also seek to fulfill the FAA's need for a consistent measure of risk assessment [47] that takes into account the effect of UAS collisions with the ground.

Other recent works seek to evaluate the operational risk of particular operations, such as urban cargo delivery by small UAS [48], which focuses on the entire risk assessment for that class of operations. "A ConOps derived UAS safety risk model" develops a risk assessment model derived from concept of operations (ConOps) that uses Bayesian Belief Networks (BBN), causal narrative, and Huggins engine to determine rolled-up probability

of failure for specific scenarios [49]. Likewise, "UAS (Unmanned Aerial System) Safety Analysis Model (USAM)" uses a BBN for developing a data-driven, integrated safety analysis model [50]. Similar work uses expert opinions to derive fault trees and associated risk analysis [51]. Combined with risk analysis frameworks discussed previously, there is fairly extensive literature for developing a risk framework and quantitative analysis for UAS. The research in this paper is not intended to replace the risk analysis frameworks developed previously, since, as already stated, using a framework such as SORA [2] or those developed by Weibel, et al. provides an adequate basis for grounding this research. Instead, this research focuses on numerical frameworks that can provide a basis for generically calculating the risks, which deals with the shortcomings of the technologies underpinning numerical frameworks already in use.

2.6.2 SORA

Since the SORA [2] framework or a similar framework is used as a basis for grounding this research, a short background of SORA is provided. More details of the SORA framework along with gap analysis can be found in [52]. SORA is derived from the JARUS guidelines document [2]. SORA [2] proposes a methodology to assess risk required to support an application for authorization to operate a UAS within the specific category. In many cases, UAS operators only desire or need to operate the UAS within a limited or restricted manner. In such cases, full design approval, airworthiness certification, type certificate, and vehicle certification consistent with a pilot's license (assuming the vehicle systems make decisions while flying) is unnecessary and restricts the ability of the UAS to perform the desired tasks. The SORA methodology is based on a bottom-up, total system safety risk assessment model that evaluates risks for a specific operation. This analysis includes all threats for the operation, the relevant design, and any mitigations to determine the boundaries for safe operation. The definition of risk used is the combination of the frequency of an occurrence and its associated level of severity. The consequence of each occurrence is

a harm. While there can be many levels of harms, multiple studies have shown that the energy associated with a crash is consistently well above the low energy levels required for a human fatality [2]. Further, human fatalities are well-defined and, in most countries, well-known by authorities. Therefore, under SORA, only human fatalities are considered as harms due to ground collisions or catastrophic mid-air collisions. As a result, the best measure of quantifying risk is the number of deaths in a given time interval or per special circumstance (such as per take-offs). The SORA bow-tie model in Figure 2.1 shows the flow from threats to harms and options for mitigations to reduce the risk.

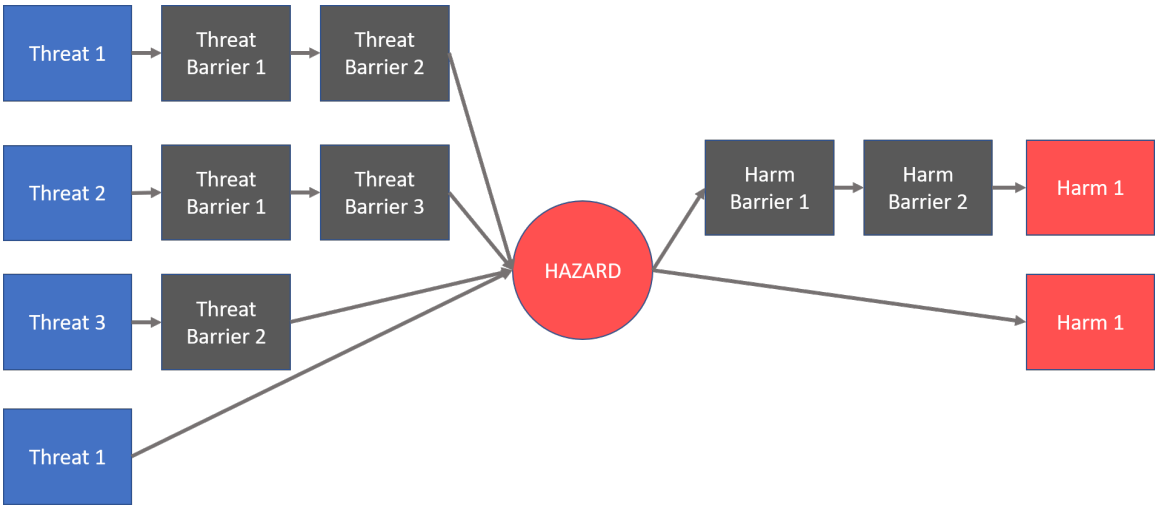

Figure 2.1: [2] The bow-tie model shows the flow from threat to harm in any specific operation. Using this model, mitigation can be employed at each step to reduce the risk that a specific threat will result in a specific harm. These mitigations are employed through risk barriers and harm barriers — steps taken to block the chain from a threat to a harm.

The three categories of harm defined in SORA are fatal injuries to third parties on the ground, fatal injuries to third parties in the air, and damage to critical infrastructure. It is the operator's responsibility to ensure that no other categories of harm arise in their specific operation. Degradation of mission is not considered a harm in SORA. The only hazard specifically defined in SORA is "UAS operation out of control" [2]. Out of control means that the operation being conducted is beyond of the approved operation, which is significantly wider than simply loss of control of the UAS. For example, the UAS entering an

unauthorized airspace, even when under full control by the operator and within visual line of sight, would be considered out of control since that airspace was not authorized for the specific operation and could lead to mid-air collision harm.

The five categories of threats identified by SORA are the following:

- Technical issue with the UAS
- Human error
- Aircraft on collision course
- Adverse operating conditions
- Deterioration of external systems supporting the UAS operation

SORA presents a framework for systematically analyzing each of these threat categories, determining the appropriate threats in each for the given specific operation, and determining the paths from those threats to the specific hazard and the three specific harms defined in SORA. These paths can be analyzed through quantitative probabilities, which are controlled by the limits in Figure 2.1, derived from an equivalence to manned aviation.

Table 2.1: [2] The probability limits of fatalities per flight hour for SORA are equal to the limits for manned aviation, since use of those limits provides a consistent basis of acceptable risk levels in aviation. These limits are further broken down to enable limiting risks depending on the category of operations.

	Number of fatal injuries to third parties on ground per flight hour	Number of hazards per flight hour	Number of persons struck per flight hour	Probability that person suffers a fatal injury
Certified Category	1^{-6}	1^{-6} to 1^{-4}	1^{-2} to >1	1
Specific Category	1^{-6}	1^{-6} to 1	1^{-5} to >1	0.01 to 1
Open Category	1^{-6}	1^{-2} to 1	1^{-5} to 1^{-2}	0 (harmless) or 0.01 to 1

The inherent difficulties associated with quantitative analysis of probabilities in complex

systems, however, limit the utility of a numerical analysis. Completeness uncertainties, due to inadequacies of the model, modeling uncertainties due to lack of knowledge of how to model complex phenomena, and parameter value uncertainties due to lack of test data to provide those values are just three of the difficulties with quantitative analyses that limit the utility of the analysis results. Therefore, SORA uses a qualitative analysis with a range of numerical levels that include risks, mitigations, and their levels of robustness. This analysis begins with an initial risk assessment. For in-air collisions, that would be an air risk assessment, with the categories shown in Figure 2.2.

	Airspace Encounter Categories (AEC)	Operational Airspace	Air Risk Class (ARC)
Integrated Airspace Operations Above 500ft	1	Class A, B, C, D, or E	4
	2	Airport environment	4
	3	Class G within Mode C Veil/TMZ	4
	4	Class G over urban environment	3
	5	Class G over rural environment	3
VLL Airspace Operations Below 500ft	6	Class A, B, C, D, or E	3
	7	Airport environment	4
	8	Class G Airspace within Mode C Veil/TMZ	3
	9	Class G Airspace over urban environment	3
	10	Class G Airspace over rural environment	2
VHL	11	Airspace above FL600	2
Any	12	Atypical Airspace	1

Figure 2.2: [2] The Air Risk Class (ARC) assessment provides the strategic risk class depending on the arena of operations, which is primarily dependent on the airspace class.

The goal through the SORA process is to ensure that the risk is commensurate with the proposed Concept of Operations (ConOps). Once the initial Air Risk Class (ARC) has been assigned, risk mitigation/reduction efforts are performed as necessary to align the risk with the proposed ConOps. These efforts are illustrated in Figure 2.3. The strategic

mitigation efforts affect the air risk class. For example, a flight plan that avoids a high-risk airspace could be substituted for a flight plan that requires passage through the high-risk airspace. Tactical mitigation efforts then follow. These efforts do not change the air risk class, but rather reduce the risk to an acceptable level within that air risk class. Depending on the level of risk reduction, there is an associated required level of robustness for the tactical mitigation effort. The final ARC and risk are then used to determine the Specific Assurance and Integrity Level (SAIL) value, which encompasses the qualitative assessment of the operational risk for all threats to the operation.

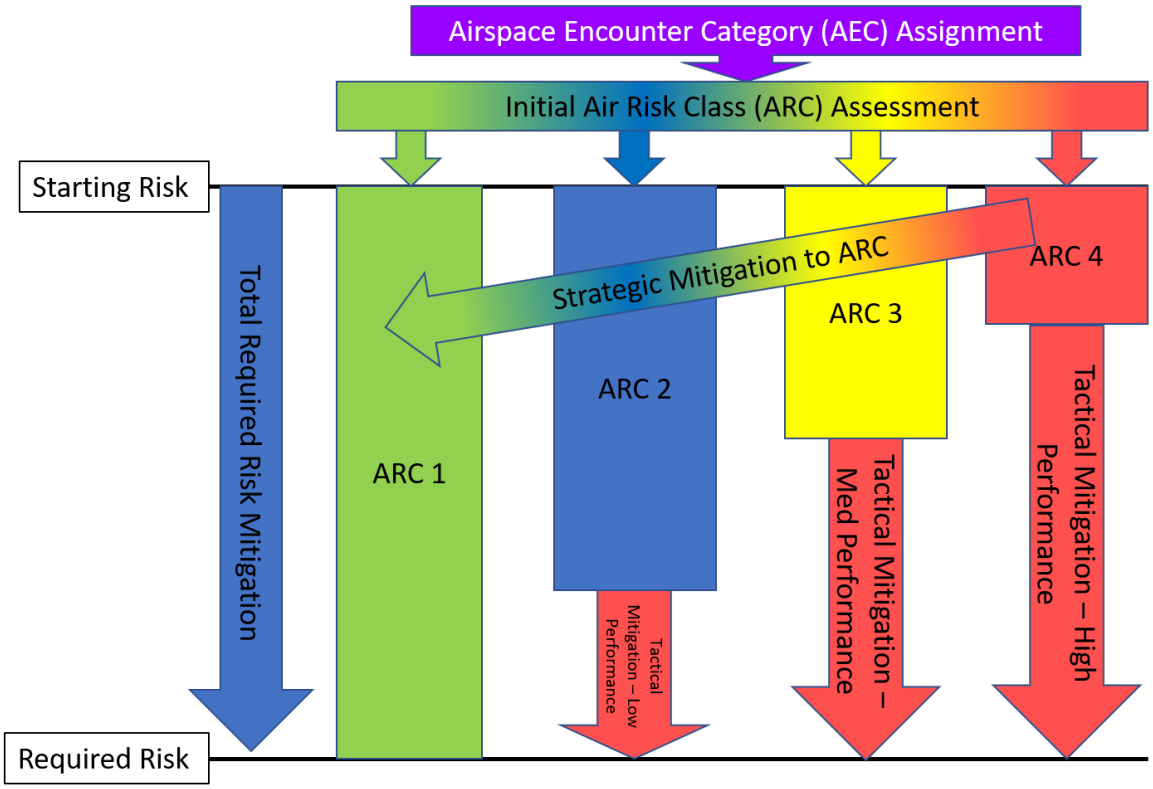

Figure 2.3: [2] This is an example risk reduction method for air encounters. Once the strategic risk assessment has been performed (the ARC assessment), tactical risk mitigation is performed to bring the risk to acceptable levels.

2.6.3 Separation Standard

In the paper "Establishing a Risk-Based Separation Standard for Unmanned Aircraft Self Separation" [53], Roland Weibel and Caroline Sieger Fernandes propose quantifying the

"well-clear" aircraft separation standard based on a time to closest point of approach analysis. Further, per the safety symposium notes in Appendix 2.1, the University of Central Florida (UCF) estimated the well-clear distance for UAS and helicopters, which went into the FAA guidance and has been corroborated via other research [54]. Further, performance standards are being defined through the Radio Technical Commission for Aviation (RTCA) for UAS in the NAS for both large and small UAS (RTCA SC-228 and the result — DO-365), including standards on detect and avoid systems [55]. Quantifying standards enables further quantifiable risk analysis, enabling a broader numerical risk analysis. The limitation, however, is that sufficient knowledge of all other air traffic that could collide with the UAS is required, through either onboard sensors or offboard information and datalinks. Each of these risk factors need to be added to the model; information that is unavailable results in higher risks and potentially unacceptable operational conditions. Since the FAA regulations that have been proposed in 2020 [11] will require transponders on all UAS, this information will be available, although datalink reliability, update rates, and other factors will still need to be handled in the risk analysis.

2.6.4 Failure Impact Analysis and Crowd Modeling

Analyzing the risk of UAS operating over populations is an essential component of integrating UAS into the NAS and has been the subject of various research endeavors [47] [56] [57] [58] [59]. Several recent research endeavors have focused on predicting the impact of a failure. For example, analysis of population centers [35] [60] [61] is used to predict the likelihood of loss of life due to a failure and resulting ground collision. Decisions based on risk analysis are comprised of two primary components: likelihood of failure and impact of failure. Without estimates of the effects of failure, the best scenario is to prevent any flight trajectories which could cause death if failure occurs. That is the current situation in which UAS are not allowed to fly over crowds or people who are not associated with the flight operations [24]. Thus, this analysis is essential to the risk analysis and decision making for

UAS. The impact of failure modeling consists of understanding the crowd or population dynamics and the vehicle dynamics during impact.

Current population modeling uses estimates based on number of people living in population centers and various models for macro-movement versus individual movement versus hybrids [62] [63]. Recommendations are made for incorporating cell phone statistics into the measures for a more real-time estimate. Since multilateration (triangulation of a transmitter position through measurements obtained via multiple receivers) is already used extensively in the cell phone industry and has been researched for developing population maps [64], centralized, anonymous statistics could be used to evaluate the safe zones for landing in the event of degradation of systems essential to a UAS [65]. While this concept works for populated areas in which there is sufficient cell phone use to anonymize the statistics, regulations prevent using this data in more rural areas in which the likelihood of identifying the cell users is too high. An alternative in rural areas is to plan flight paths via zip codes, using population statistics for the zip codes, provided the UAS has sufficient range. A third method for capturing crowds in real-time is to use geo-located social media feeds. For example, Twitter geo-located tweets can be used for analysis of crowd formation and movement [66]. Using video feeds from UAS, research has also shown that neural networks can be used to improve the crowd modeling, thereby supplementing data sources with direct observations [67].

While the previous research focused on population and crowd modeling while including basic dynamics of the vehicle upon impact, other research has been done to model the dynamics of the vehicle including bounce effects to better understand the potentially affected area [68].

2.6.5 Path Planning

A significant portion of handling off-nominal conditions is path planning and estimation, both before and after events occur. Path planning before the occurrence of an event that causes off-nominal conditions has the ability to provide a safety net of options in the case of such an event. This type of path planning, based on nominal vehicle dynamics, is well researched, with further ongoing investigations. These planners typically perform optimizations with parameters that allow the designer to choose the balance between safety (related to particular situations) and mission efficiency. A good example of this type of path planning is detailed in the work by Vian, et al. [69]. Once an event has occurred that causes off-nominal conditions, the flight trajectory prediction is based on the new vehicle dynamics. Since the off-nominal vehicle dynamics are not known *a-priori*, they must be estimated either based on known failures or through an online learning algorithm. For example, the neural network capabilities in the GUST software [70] enable learning the new vehicle dynamics online, allowing path planning and estimation to be based on the updated model of the vehicle dynamics [71] [72] [73]. These path planner may either be short or long time horizon planners depending on the requirements for the contingency reaction.

2.6.6 Fault Detection, Identification, and Accommodation (FDIA)

For small UAS, fault detection is often difficult as there are not always backup or monitoring systems available to detect or handle subsystem failures. In the case of the multirotor, for example, the failure of the IMU can be catastrophic since stabilization is required for multirotors, and the IMU is required for stabilization. Newer autopilot hardware often includes more than one IMU, but even a second IMU only provides a backup; it does not provide fault detection capabilities. Redundancies and detection for many aspects of small UAS, including IMU, can be provided if designed properly [74]. An option for IMU FDIA is proposed in “Fault Detection, Identification and Accommodation Techniques for Un-

manned Airborne Vehicle" by Lennon R. Cork, Rodney Walker, and Shane Dunn [12]. In their research, they proposed using a neural network to detect abnormal operation of the IMU and to replace the IMU data with output from the neural network should such a detection occur. While neural networks are limited in application for flight approval since it is difficult to prove that neural networks have repeatable responses, this concept demonstrates that even the most central of sensors for multirotor stabilization can be provided with both a backup and observer. These methods provide an information input to a risk analysis model — both as a safety alternative and the probability of the alternative fulfilling its role should the primary system fail. FDIA techniques enable lower risk of primary system failure since detection and accommodation can potentially handle the first failure, allowing the vehicle to land or divert safely within operational parameters on the backup system.

Fixed wing aircraft provide the easiest backup systems in the event of motor failure since they can glide safely to a landing, should a landing zone be available. Options provided by Pedro Fernando Almeida Di Donato in his PhD dissertation "Toward Autonomous Aircraft Emergency Landing Planning" at the University of Michigan in 2017 enable potentially safe landing options for UAS [13]. Further, autonomous auto-rotation for rotary wing aircraft, such as provided in "Flight Path Planning for Descent-phase Helicopter Autorotation" by Thanan Yomchinda, Joseph F. Horn, and Jack W. Langelaan [75] or by a Rockwell Collins patent [76] enables safe descent methodologies for the third major class of UAS (rotary wing versus plane or multirotor). These technologies enable online risk analysis with backup options. However, there still remains the risk associated with whether the FDIA system correctly handles these faults, and more importantly for risk analysis, whether that risk is quantifiable. This question goes back to the heart of system design and development with a focus on the approval process. If FDIA systems cannot adequately quantify their ability to detect and handle faults, then the risk analysis must ultimately still be handled by an *a-priori* qualitative risk analysis, which limits the ability of these systems to make

decisions in real-time.

A recent development regarding FDIA is the use of a backtracking algorithm to determine the critical failures that can cause a specific degradation or failure to occur [77]. In addition to FDIA pathways, this method also provides risk assessment of the critical failures and probability of the downstream failure to occur, enabling this technology to be easily included in risk assessment frameworks that use probabilities as the basis of the risk assessment.

2.6.7 Online Risk Analysis

There is another option for a backup system to handle contingencies: predict the effect of off-nominal behavior on the trajectory of the UAS. There is a subtle difference between this option (online risk analysis) and the previous option (FDIA). In the previous option, the backup system must detect a failure (fault identification) and handle it (fault handling) in order to return the system to a near-nominal condition. This subject has already been and continues to be researched extensively. Online risk analysis leaves detection and handling of failure to other systems such as FDIA. Instead, online risk analysis stochastically predicts the new trajectory of the UAS [14], enabling either an operator or an intelligent system to respond appropriately. In other words, online risk analysis tells the operator what will likely happen as a result of the fault, without attempting to handle the fault. This option can only be performed online since the cause of an off-nominal trajectory is not known until it occurs. Online risk analysis provides several advantages, but the primary one is the ability to delay decisions until they become necessary. For example, a system without any online analysis must restrict its flight to ensure all contingencies do not cause out of control operations. Online risk analysis provides the means to allow a broader flight envelope until the risk of out of control operations is too great. Flight trajectory predictions provide the operator with a means to steer the trajectory to minimize the impact of the off-nominal

conditions. Further, these evaluations do not need to occur in real-time. Close to real-time is sufficient, provided that the time delay of computations is incorporated into the trajectory predictions.

A couple papers have recently been published which begin to realize this option. "Real-time Risk Assessment Framework for Unmanned Aircraft System (UAS) Traffic Management (UTM)" by Ancel, et al. [14] provides the first real-time risk assessment implementation in literature for a UAS. This is a significant achievement towards the realization of real-time risk analysis and adaptation for UAS, but there are still many gaps to be addressed. The primary gaps in this research are the use of offline resources for computations, which become unusable in the event of loss of communications, and the restriction to Bayesian Belief Networks, which typically require a reasonable set of *a-priori* data to provide the priors. Likewise, recent developments have been made in online guidance updates based on collision risk assessment [78]. This research uses decision trees and probability of collision to determine whether to maneuver and, if so, which policy choices to follow based on offline learning.

In "A Unified Approach to Separation Assurance and Collision Avoidance for UAS Operations and Traffic Management", Ramasamy, et al. develops a rigorous analysis of air vehicle avoidance volumes using the errors built up in the navigation system [15]. Specifically, this approach both provides the avoidance volumes for a given system and allows the computations of the necessary avionics to achieve a desired avoidance volume. This work, together with the former risk assessment framework, provides a mechanism for real-time evaluation of a vehicle's trajectory and the areas it might impact. Further, combining these analyses with the failure impact analysis in Section 2.6.4 provides a complete means to evaluate the air or ground collision risk of a UAS, albeit with some shortcomings.

2.6.8 Dempster-Shafer Theory

In general, quantitative risk analysis frameworks require an underlying technology to calculate the risk, usually in the form of probabilities. The frameworks discussed in Section 2.6.1 use various methods; Bayesian Belief Networks is used several times as a structured data method that allows reasoning over stochastic variables. A less well-known theory which applies to this problem is Dempster-Shafer Theory, which many view as a generalization of Bayesian reasoning [3]. However, unlike Bayesian reasoning, Dempster-Shafer Theory does not require knowledge of priors. Furthermore, Dempster-Shafer is based on the concept of probabilities over sets of sets, including the complete set, which is equivalent to unknown information. Thus, Dempster-Shafer explicitly defines and calculates what portion of the probability distribution is unknown, allowing the theory to start from fully unknown data (i.e. no priors) as well as make different decisions when faced with lack of information versus balanced probabilities between various options. Primarily used in sensor fusion and risk analysis [79] and typically requiring higher computational resources, Dempster-Shafer Theory provides useful properties that overcome some of the issues facing the risk calculation technologies underpinning quantitative, general risk analysis frameworks discussed in Section 2.6.1. Chapter 3 discusses in-depth background research in Dempster-Shafer Theory and develops extensions which apply to the UAS risk analysis problem discussed here.

CHAPTER 3

DEMPSTER-SHAFER RISK ANALYSIS FRAMEWORK

3.1 Introduction to Numerical Risk Analysis Frameworks for Unmanned Systems

Central to the concept of a risk-based assessment for Unmanned Aerial Systems (UASs) operations and the ecosystem that surrounds that assessment is a mathematical framework capable of computing the risk. There are many frameworks currently available for this type of risk computation. One of the most basic is a rules-based analysis, commonly known as "expert systems" [80]. While easy to understand and explain the results, these systems suffer from being inflexible, requiring branches to be updated as new information and options are revealed. Further, risk analysis inherently requires capturing stochastic distributions, for which expert systems are not well-suited. A second option, case-based reasoning, also provides a methodology which is easy to explain and relates well to human decision-making [81]. This method can better capture stochastic distributions and mirrors the concept of SORA which builds cases for each flight authorization and adds those cases to a library to be building blocks for future authorization requests. However, this method suffers from a need for each case to be analyzed by a human to choose the important features before the case is entered into a library for future analyses, limiting the flexibility of this method. Moving away from cognitive methods, data-driven methods such as neural networks [36], Gaussian processes [82], and support vector machines [83] offer the flexibility of adapting to new information as it becomes available. Further, data-driven methods easily capture the stochastic distributions required to analyze risk. The three principle limitations to apply

these methods to UAS risk analysis are the following:

- Lack of information
- Lack of computing power
- Lack of explainability

Currently, limited information is available for risk analysis of UAS, as discussed in the 2018 Safety Symposium notes in Appendix F. New UAS products and upgrades from companies such as DJI [7] are often available, departing from a more traditional manned aviation model in which vehicles product updates are slower, allowing for an improved understanding of the risks associated with operation in the NAS. As such, the mathematical analysis framework will need to be able to start from nearly no quantitative evidence and build over time to incorporate new data from subsequent operations. Purely data-driven methods require substantial training data before they can provide usable results [36]. Further, purely data-driven methods typically require significantly higher computing power to run the analysis since relationships between data points are calculated rather than given by expert input. Since one of the goals of the risk analysis framework is to run on autopilots onboard UAS, an ability to run on embedded systems with lower computing power is essential. Finally, the conclusions from purely data-driven methods are much more difficult to explain. For a system which operates under human oversight, such as the US national airspace operating under the oversight of the Federal Aviation Administration (FAA), it is necessary for the reasoning behind decisions to be explainable to the humans providing oversight to the system. This condition makes purely data-driven methods difficult to use in the NAS.

A third category of quantitative risk analysis frameworks is structured data. This category is a blend of the previous two, combining the knowledge of subject matter experts (SMEs) for structuring the data while enabling sufficient data flexibility to encompass stochastic distributions and risk analysis. Methods in this category include Bayesian reasoning and Markov processes. Due to their data structure, these methods are sufficiently explainable

to serve as a basis for decisions in a system with human oversight. Further, efficient means of computing decisions through these networks are available [84].

Due to the afore-mentioned method evaluation, the numerical framework initially chosen for this risk-analysis was a Bayesian network. However, Bayesian networks still run into their own limitations in that these networks require probability distributions to initialize the network. Typically, the data and time requirements to obtain the initial probability distributions are quite high. Since UAS are still in their infancy, the initial probability distributions are, in general, unknown. This is also true because the capabilities and designs of UAS are constantly changing without presenting enough information about the ever-new systems to understand the risk associated with them. Thus, the chosen mathematical framework must incorporate the capacity to start from unknown information and gradually incorporate sufficient information to provide informed risk assessments. Evidential reasoning, also named Dempster-Shafer theory after the original developers of the theory [1] [3], is a structured data approach similar to Bayesian. However, Dempster-Shafer theory quantifies "unknown" data — the component of the probabilities that could be any stated option. By quantifying unknown data, Dempster-Shafer theory enables starting an analysis with no *a-priori* knowledge of the probability distribution.

This section is structured as follows:

- Section 3.2 provides a brief overview of Dempster-Shafer theory
- Section 3.2.3 provides a brief overview of evidence propagation through a network
- Section 3.3.1 discusses the requirements of this network to combine evidence at each node
- Section 3.3.2 develops novel rules to update the transition potential matrices based on evidence inputs to nodes
- Section 3.4 develops the novel use of episodic learning for Dempster-Shafer net-

works, improving the transition update results

- Section 3.5 develops novel rules for evidential weighting to combine evidence in the network
- Section 3.6 shows tests and results for the novel update rules
- Section 3.7 discusses the conclusions, limitations, and future work for the novel transition potential matrix update methods

3.2 Dempster-Shafer Theory

Dempster-Shafer (DS) theory was originally devised by Arthur Dempster [1] and Glenn Shafer [3]. The following papers provide a mathematical background to the theory [79] [85]. "Smart Projectile State Estimation Using Evidence Theory" provides an practical understanding of evidence theory using sensor fusion and state estimation as the backdrop [86]. Other practical explanations of DS theory are available [87]. For the purpose of this work, focusing on extensions to DS theory as applied to networks, this chapter will start with a simple, practical example to set the stage for understanding what the DS network offers. This example will be related to Bayesian reasoning for readers familiar with that framework. Readers who are already familiar with DS theory and its complications can jump to Section 3.2.3.

3.2.1 Dempster-Shafer Information Fusion Example

Using the nomenclature previously defined, this section focuses on a simple example which will make the DS theory and application clearer. Consider a situation in which an object is one of the following options as shown in Figure 3.1.

- Red ball
- Green ball

- Red cube

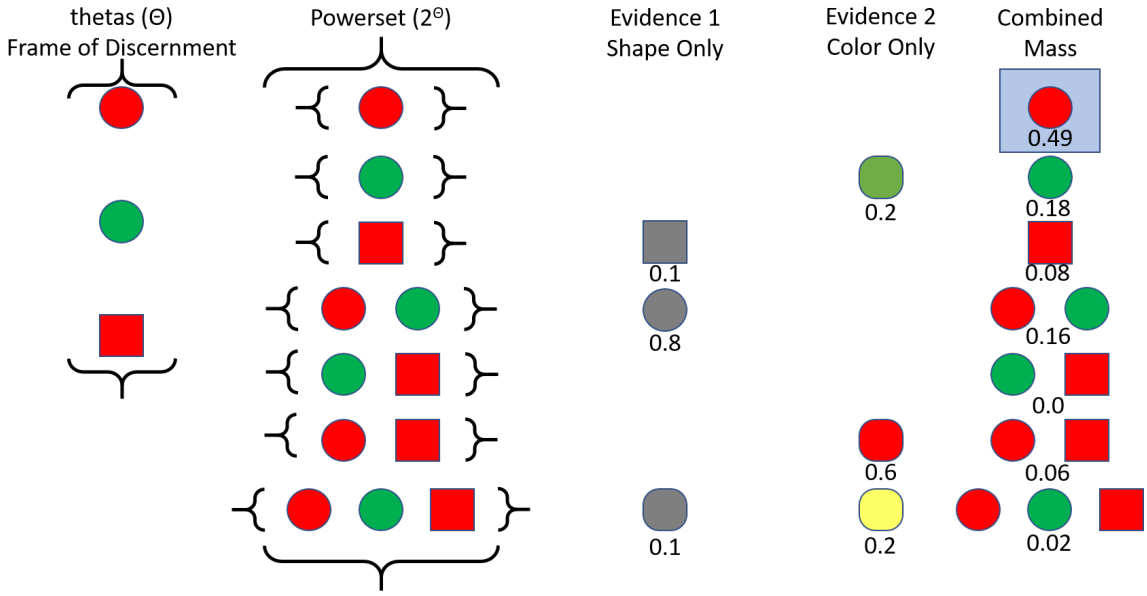

Figure 3.1: Visual representation of the simple Dempster-Shafer example. Θ represents the three options that could be observed by the sensors. The powerset column represents all combinations which Dempster-Shafer analysis considers. Recall that sets with multiple options signifies the belief that the observed object could be any one of the objects in the set. The two sensors that provide observations — evidence — can detect all objects, but can only detect certain properties of each object. Those detections are shown, along with the belief masses assigned to each element of the powerset: the Basic Probability Assignment (BPA). The combined mass column shows the powerset again, along with the results of the analysis which correspond to the greater details in Table 3.1. The highlighted element of the powerset (the red ball) is the θ which is believed to correspond to the true object, based on the Dempster-Shafer analysis.

Together, these options comprise the Frame of Discernment, Θ, as the object under question can only be one of these. The powerset of Θ is then shown in the "Powerset" column of Table 3.1. Assume two sensors provide evidence concerning the object. Evidence one is provided by a black and white camera that can only distinguish shape but with error. Evidence two is provided by a sensor that only distinguishes color with error. Suppose the object in question is a red ball. The evidence provided by sensor one may look similar to the belief masses in the "Evidence 1" of Table 3.1. The "Evidence 1" column is then a BPA assigning the belief masses to each element in the powerset. All elements

with non-zero mass are Focal Points. This evidence can be interpreted as the sensor is 10% sure the object is a cube, 80% sure the object is a ball, and 10% unsure of what the object is.

The "Evidence 2" column represents a potential set of evidence from a sensor that only distinguishes color. In this case, sensor two is 20% sure the object is green, 60% sure the object is red, and 20% unsure of the color. This sensor is less precise at distinguishing colors than sensor one is at distinguishing shapes. Since these two evidence sets are not in high conflict, which will be discussed in more detail in Section 3.2.2, Dempster's Rule [1] can be used to combine the evidence, and the result is shown in the "Combined" column of Table 3.1.

Table 3.1: Dempster-Shafer Evidence Example. The "Powerset" column represents the full set of options to which a BPA can be assigned. The "Evidence 1" column shows the first evidence set from the sensor that distinguishes shape. The "'Evidence 2" column shows a second evidence from a sensor that distinguishes color. The "Combined" column shows the rounded, combined masses based on Dempster's Rule, and the "Bel" and "Pl" columns show the Belief and Plausibility functions, respectively, for each of the elements of the powerset of the combined data.

Powerset	Evidence 1	Evidence 2	Combined	Bel	Pl
Red ball	0.0	0.0	0.490	0.490	0.734
Green ball	0.0	0.2	0.184	0.184	0.367
Red cube	0.1	0.0	0.082	0.082	0.163
(Red ball, Green ball)	0.8	0.0	0.163	0.837	0.857
(Green ball, Red cube)	0.0	0.0	0.000	0.266	0.286
(Red ball, Red cube)	0.0	0.6	0.061	0.633	0.653
(Red ball, Green ball, Red cube)	0.1	0.2	0.020	1.0	1.0

First, note that only the subset green ball and red cube had zero evidence assigned from either sensor. All other subsets either had an ambiguity or a θ that was a focal point of each evidence set. Dempster's Rule follows the single-vote-no concept in that, if a single evidence set assigns zero mass to an element of the powerset and to any set that contains that element, then that element will be eliminated from the combined result. Second, notice that the unknown element — the complete set — is significantly reduced in the combined dataset from each of the two evidence sets. Since the evidence sets were not highly con-

flicted, unknowns and ambiguities (sets that include more than one θ but not the complete set) were reduced. Finally, note that the correct classification, the red ball, has the highest combined mass of any of the elements of the powerset. Looking at the "Bel" and "Pl" columns of Table 3.1 — the belief and plausibility functions, respectively — one can see that the belief functions are the sums of all masses that could apply to that element, and the plausibility functions are one minus the sum of all masses that could not apply to that element. Thus, the belief function for red ball equals the combined mass for red ball while the belief function for red ball and green ball is equal to the sum of the belief masses for red ball, green ball, and (red ball, green ball). Likewise, the plausibility function for red ball is one minus the sum of the belief masses for green ball, red cube, and (green ball, red cube). Looking at the difference between the belief function and plausibility function columns leads to a few conclusions:

- This example concludes that there is a precise understanding of the belief associated with the object being either a red ball or a red cube because the belief and plausibility function values — the "Bel" and "Pl" values for the (Red ball, Red cube) row in Table 3.1, which represent the lower and upper bounds on the belief — are nearly the same value.

- There is a similar precise understanding of the belief associated with the object being either a red ball or a green ball, but the belief in that case is significantly higher than the belief that the object is either a red ball or a red cube. This result is expected since we know from the example setup that the sensor that distinguishes shapes is more precise than the sensor that distinguishes color.

- The largest range of belief values is associated with the correct object classification – the red ball. This classification is also the strongest belief and highest plausibility of any of the θ elements.

- If a decision-maker were to play it safe, the decision-make could state that it is most

likely that the object is either a red ball or a green ball. If the decision-maker were willing to accept a bit more risk, then that decision-maker could state that the object is likely a red ball. The choice the decision-make will make is governed by the acceptable limits placed on the belief and plausibility functions.

Finally, we take a look at this same problem from a Bayesian perspective [88]. Before doing this, we must note one nuance concerning the DS approach. The DS approach had an original unstated hypothesis concerning the object: that the type of object was unknown (all belief mass is assigned to the complete set). When Dempster's rule is used to combine a BPA of all mass assigned to the complete set with another BPA, the result is the same as the second BPA, thus allowing that step to be removed from the simple example. For a comparable Bayesian example, we start with all mass equally divided among the Θ elements, using similar terminology as the DS example for ease of comparison. The first point to note is that the closest representation of unknown in Bayesian is equal probability distribution across all options. However, this distribution is indistinguishable between equal probabilities of all options being correct versus no knowledge of which option is correct. The Bayesian observations are then cube versus ball for the first evidence and red versus green for the second evidence. The ambiguities in the DS evidence translate into likelihoods (correct and incorrect) for the Bayesian test, and we assume the observation is correct — a red ball. The result of these computations is shown in Table 3.2. From this analysis, we can make a few observations:

- The correct answer — red ball — has the highest posterior probability (the final column in Table 3.2), meaning that the object in question is most likely a red ball.
- The evidence was highly supportive of the correct answer to the extent in the DS example that the belief and plausibility functions of the Θ elements don't overlap. The Bayesian result does not communicate this information at all, meaning that in the Bayesian analysis, the decision-maker cannot be as confident in the decision that

Table 3.2: Bayesian probability Example. This example mirrors the DS example in Table 3.1 as closely as possible for comparison. Because the evidence sets are direct observations of the priors, the likelihood is 1.0.

Θ	Prior	Likelihood 1	Posterior	Likelihood 2	Posterior
Red ball	0.333		0.459		0.623
Green ball	0.333		0.459		0.267
Red cube	0.333		0.081		0.110
incorrect shape		0.15			
correct shape		0.85			
incorrect color				0.3	
correct color				0.7	

the object is a red ball.

In summary, this simple example resulted in the same outcome with Bayesian and DS analyses. However, the additional information contained in the DS analysis presents the reliability of the result to the decision-maker. For a decision based on the highest probability, either analysis works. However, suppose for example that the decision-maker wants to ensure that the object in question is not a red cube or a green ball. In the Bayesian analysis, there is no way to know the upper limit on the probability for the object either being a red cube or a green ball. From Table 3.1, in the DS analysis, the upper limit of the probability that the object is a green ball is 0.367. The upper limit on the probability that the object is a red cube is 0.163. The lower limit that the object is a red ball is 0.490. Thus, for a decision-maker whose responsibility is ensuring that the object is not a red cube or green ball, the DS analysis provides the information necessary to make that decision. The Bayesian analysis does not provide the required information. This is one reason why DS analysis is used heavily in risk analysis and sensor fusion, typically for classification of sensed objects.

3.2.2 Dempster-Shafer Combination Rules

With this basis in understanding of the application of DS analysis, the next step is to look at the combination rules. The original DS combination rule is linear, which preserves the axioms required for probability combinations [1], thus allowing this method to reduce to Bayesian reasoning under certain conditions. However, this rule suffers from a couple of issues, the principle one being the assumption that all evidence contributors have equal weight, reliability, and full knowledge of the frame of discernment. In our simple example in Section 3.2.1, each sensor still had full knowledge of all elements of the powerset (can detect every possible combination), but returned ambiguities when it could not distinguish between specific θs. This application adheres to the assumptions of Dempster's Rule. In contrast, suppose that sensor one could not detect red cubes. In that case, sensor one would always assign zero mass to any subset of the powerset that contains red cubes. This situation violates the assumptions of Dempster's Rule and leads to non-intuitive results due to the classic vote-no-by-one issue in which a single no-vote by an evidence contributor results in the option being assigned a belief of zero when the evidence is combined. For highly conflicting data, this results in non-intuitive results such as in Table 3.3 [79].

Table 3.3: Dempster-Shafer Conflicting Example. The combination of highly conflicting data provides non-intuitive results. In this case, although both A and C each have a large belief mass in an evidence set, 0 mass for each of A and C in the other evidence set results in a vote-no-by-one scenario in which one sensor "votes no" for A and the other sensor "votes no" for C. The result that all belief mass is given to B when combined. Note that for this simple example, only single options are focal points in the frame of discernment.

Data Set	A	B	C	(A,B)	(A,C)	(B,C)	(A,B,C)
Evidence 1	0.9	0.1	0.0	0.0	0.0	0.0	0.0
Evidence 2	0.0	0.1	0.9	0.0	0.0	0.0	0.0
Combination	0.0	1.0	0.0	0.0	0.0	0.0	0.0

Because the assumptions of Dempster's Rule are often not applicable to real-life scenarios (in most situations, all evidence observations do not have full knowledge of all elements of the powerset and are not equally reliable), multiple authors since the 1980s have devised

ways around these assumptions including different combination rules as well as different input functions that add unknown mass to account for the lack of knowledge of elements of the powerset. Since this paper is not an overview of these combination rules, the focus will be placed on three combination rules that have useful properties for the risk analysis being developed. Zhang's combination method [18], Murphy's combination method [19], and the Evidential Reasoning rule [89] are three such methods that enable a custom weighting to be associated with each new evidence. This property will be useful later in the development of the risk analysis network in Section 3.5.

3.2.3 Evidence Propagation

Evidence propagation through a hypertree [17] was first introduced in the 1980s. Similar to Bayesian propagation, the principle difference is that evidence propagation is less concerned with the transition values being conditional probabilities. This is best shown through Figure 3.2, which shows a parent node with its transition to a child node. Note that it is easy to convert transition values into conditional probabilities [17] [4]. Since DS reasoning condenses down to Bayesian reasoning under certain assumptions [90], evidence propagation is very similar to Bayesian propagation. In fact, the example given in the original work by Shafer and Shenoy is a Bayesian example that overwrites nodes with new evidence and treats conditional probabilities at each transition as a known fact used for evidence propagation [17]. In order to maintain consistency in the network with known conditional probabilities, it is necessary to overwrite node information with marginal probabilities inferred through the conditional transitions based on the most recent evidence update. The effect of this can also be seen through Figure 3.2 since propagating even an identical input to the previous child marginals does not result in the parent marginals being recovered. This mechanism assumes that the most recent evidence is the best choice and, therefore, limits the capabilities for the network to incorporate uncertain evidence at each node.

This original work was extended through the 1990s and early 2000s under various names

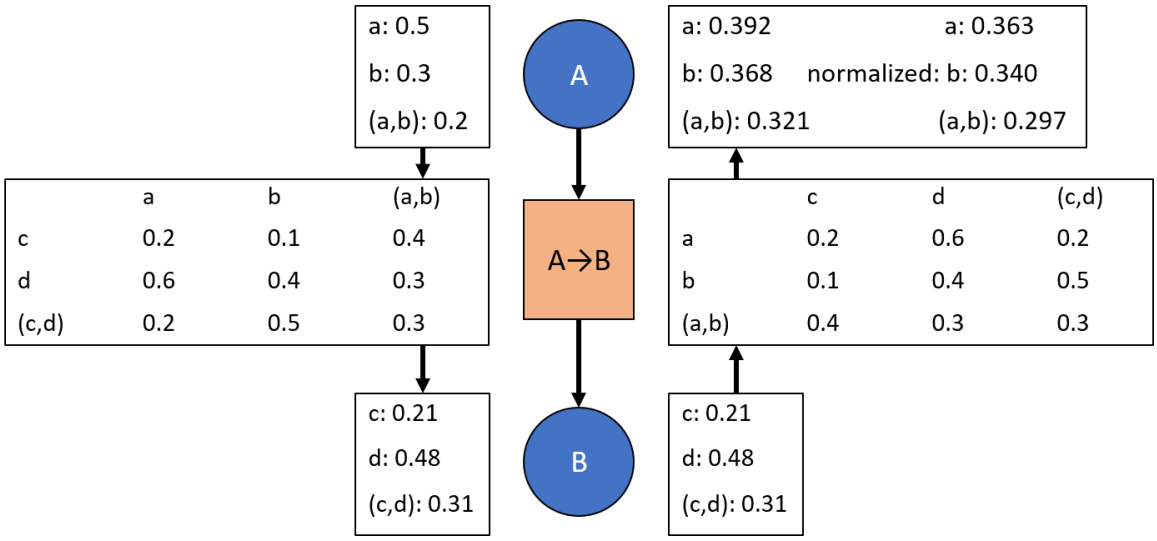

Figure 3.2: Propagating the Dempster-Shafer evidence masses through the transition between nodes. For simplicity, only two options are available at each node. The direction of propagation is represented by the arrows between evidence masses. The arrows between the nodes represent the definition of the network. As can be seen, evidence propagation from node A to node B results in normalized masses. Conversely, evidence propagation from node B to node A results in non-normalized masses. Moreover, the original masses are not recovered if masses are propagated from node A to node B to node A through the same transition.

for the field of study. Valuation networks is the general name given to these networks by Shenoy [16], which are not required to be directed (i.e. each connection between nodes has a direction associated with it) or acyclic (i.e. given any starting node, there are no paths in the network following the directions between nodes that return to the starting node) and encode the relationships between characterizations of the uncertainty for local sets of knowledge. The determination of whether sets of knowledge can be broken into separate nodes is based on conditional independence [91], e.g. if sets of knowledge are conditionally independent from each other, then they can be broken into separate nodes with the relationship encoded on the link between the nodes. Uncertainty propagation was extended by Smets [4] [92] based on the Transferable Belief Model (TBM). While the Transferable Belief Model is a powerful tool for capturing the relationships between knowledge sets, it is based on the non-probabilistic belief function theory [93], in contrast to the probability-based belief function theory that underpins DS Theory [93]. Using the formal theory of

TBM, Smets showed the relationships between the joint probabilities used in Shenoy's original work [17] and conditional probabilities used in Bayesian networks [4]. Specifically, Smets showed that conditional probabilities for each of the θs in node A that are conditionally dependent on node B could fully capture the effects of the probability distribution of node B on the θs in node A [4]. This relationship allowed the joint probabilities from Shafer and Shenoy [17] to be reduced to the minimum representation of conditional probabilities, which results in reduced computer memory usage and could leverage conditional probability calculations already developed for Bayesian networks. This work applies to valuation networks in general — no directed acyclic assumptions required — and has since been extended [94].

More recently, primarily in the 2000s, work has returned to the joint probabilities originally used by Shafer and Shenoy [17] for encoding the relationships between nodes in a valuation network. Evidential networks — valuation networks which use evidential reasoning to combine observations at nodes and joint mass tables (the general case of joint probabilities as seen in Figure 3.2) — have been extended and applied to various scenarios including threat assessment [93] [95] [96]. Further, discounting has been introduced to reduce the weight of inferred evidence versus directly observed evidence [95]. These extensions and applications are consistent with the original work [17] while offering improvements in computation speed, reliability weighting of evidence, and analysis of what evidence can be propagated between nodes. However, these extensions are still based on the same assumptions about joint masses — the joint masses are set and updated occasionally by someone who has knowledge of the relationships between nodes. This concept of unchanging joint masses or joint probabilities is an assumption that allows the joint probabilities to be represented as conditional probabilities. This assumption clearly underlies the use of vector projections and spans to show that the conditional probabilities are a minimal representation (i.e. minimum information required) of the joint probabilities [4]. Consider

the case in Figure 3.3 in which variable d is not influenced by the variables in node A, thus allowing the conditional probability representation to be smaller than the joint probability representation.

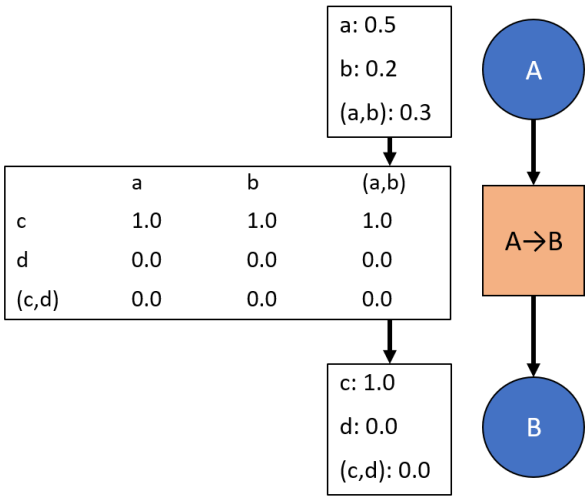

Figure 3.3: An example case in which the conditional probabilities are a minimum, and a smaller, representation of the joint probabilities. The variable d in node B is not influenced at all by the variables a, and b, in Node A and is, thus, independent from the variables in Node A. The conditional probabilities would reflect that by eliminating that row of zeros, resulting in fewer values being stored to represent the relationship.

While this assumption is useful if the joint probabilities are not changing often, the application of the evidential network in this paper is to unmanned systems, and an underlying assumption of this analysis is that the relationships between nodes are constantly changing as operational data is received. Thus, the move from joint probabilities to conditional probabilities actually causes an issue.

Published papers that are based on evidential reasoning networks often assume a directed acyclic network, such as the work by Pollard and Pannetier [95], which can model the knowledge required for the applications described in this paper. Further, directed acyclic networks with multi-path loops can always be reconstructed to remove the loops, as shown in Figure 3.4. Since directed acyclic networks are sufficient for the decision analysis in this paper, directed acyclic networks are assumed for the rest of the developments in this

paper.

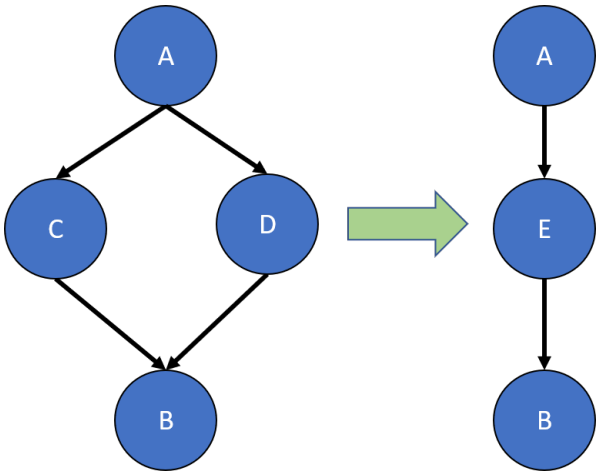

Figure 3.4: A simple example of how to remove multi-path in a directed acyclic graph. The variables in nodes C and D are combined into node E, resulting in a single path from A to B. This combination is always possible since belief masses assigned to sets including variables from both nodes D and D can always be set to zero, thereby resulting in two independent belief mass distributions contained within the same node.

Other methods of handling valuation network representations have been developed [97] [98]. While these representations have less in common with the evidential reasoning networks, limitations are introduced to prevent information from being incorrectly inferred [97]. For example, if the network is comprised of two nodes such that the parent node has two options: "writer" and "not writer", and the child node has two options: "journalist" and "not journalist", evidence suggesting the entity under question is not a journalist cannot be used to infer evidence against the entity being a writer.

3.3 New Rules for Evidence Propagation

Current rules for updating the network have difficulties with highly limited *a-priori* data because the transition potentials are only updated based on user inputs, thus requiring sufficient knowledge of the relationships between nodes before the network is used. New rules have been developed to facilitate intuitive evidence propagation when transitions between nodes start with limited or unknown information. Unknown transition information

is represented as transitions mapping all marginal masses to the complete set as shown in Figure 3.5. This state is referred to as "vacuous" [4]: providing no information on the joint probabilities between the nodes.

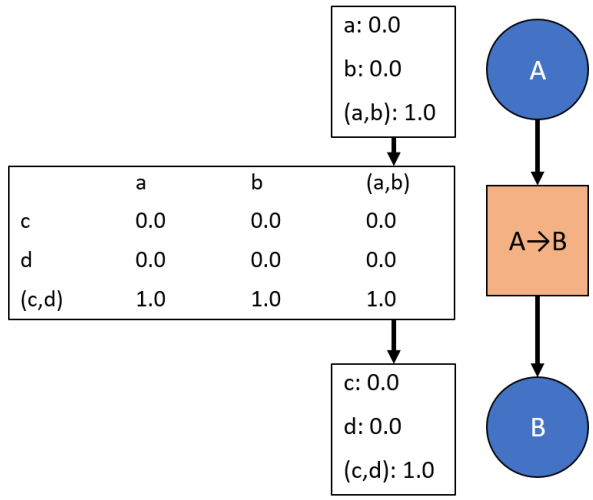

Figure 3.5: An example network that starts with no data. This analysis is beyond the scope of Bayesian logic since Bayesian requires priors. Further, overwriting information is risky in this context since a full overwrite of information suggests sufficient data behind each update. In other words, the first overwrite would be similar to starting with Bayesian logic after the first update, but insufficient information for that to occur has already been assumed.

3.3.1 Evidence Combination at Nodes

One of the principle strengths of DS theory is combining evidence in a single frame of discernment. Developing a hypertree or network enables structuring the data such that each local frame of discernment is computationally feasible. For example, as in Figure 3.2, each node has 2 options, which means each powerset comprises 3 options (see Nomenclature for powerset definition). Without the structure, there are 4 options available, which means a powerset of 10. As the number of nodes and hypotheses (θs) per node grow, an evidential reasoning network quickly becomes the only computationally feasible option for limited computing power. In addition, information about causality between hypotheses is obfuscated. Within a network, however, new evidence provided at each node often still in-

cludes high uncertainty. Further, the newest evidence is not necessarily the most reliable — i.e. overwriting data at each node with new information is not necessarily the best update mechanism. As a simple example, new data for aircraft engine failures over a period of time since the last update does not mean that the new probabilities are the best to use, especially if the engines have not been modified since the previous period. It may be better to consider the sets of evidence equally and combine evidence at that node. This is especially true for evidence inferred through a transition from another node, which can be considered to be less reliable than directly observed evidence.

The first rule defined for this network is that all evidence updates at each node uses an evidential combination rule as shown in Figure 3.6. This rule aligns with previous work [95] [99]. In Section 3.2 it was noted that there are multiple combination rules that have been developed as alternatives to handle the non-intuitive results of conflict in the DS rule. Some rules, such as Yager's method, are dependent upon the order in which data is entered [20]. These rules create an intrinsic weighting of evidence based on temporal order. This distinction is important since the weighting discussed in Section 3.5 does not need to handle temporal order. The inherent drawback to using combination methods to update each node is immediately realized in that the network values are no longer consistent with each other once evidence has been propagated through a transition and combined at a node — i.e. the node's marginal probabilities no longer equal the previous node's marginal probabilities multiplied by the transitional values, which can be clearly seen in Figure 3.6. This result suggests the possibility of learning the transitional values through updating the transitions to be consistent with the new node values. This update is discussed in Section 3.3.2.

3.3.2 Transition Updates

As stated in Section 3.3.1, subsequent to combining evidence at a node, the network is no longer locally consistent across transitions to the neighboring nodes. This state leads to the

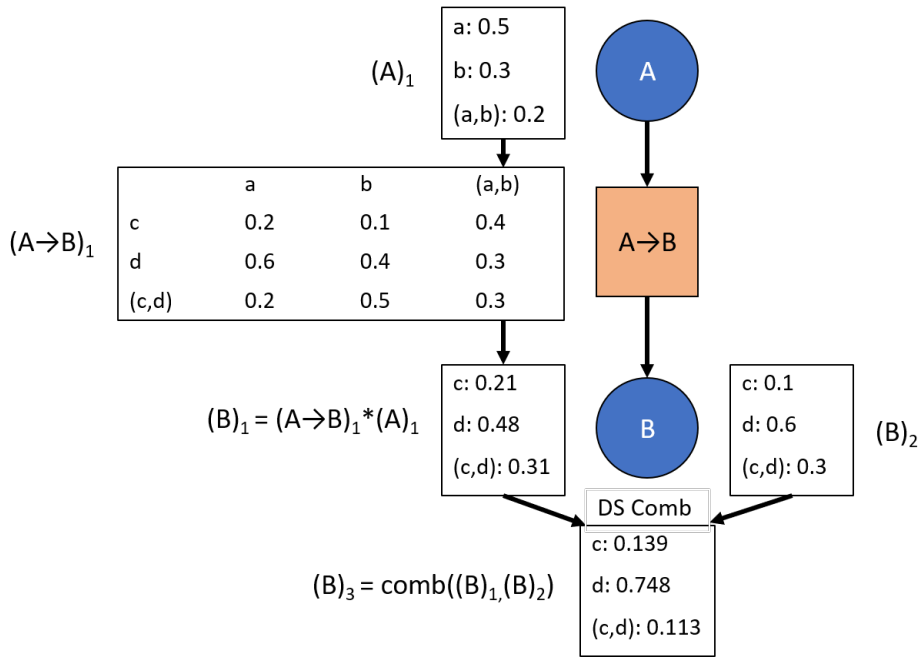

Figure 3.6: An example network update that uses Dempster-Shafer updates to combine evidence at each node. The combined value from node A $(A)_1$ is propagated as evidence through the transition to node B $(B)_1$, where it is combined with directly injected evidence $(B)_2$, resulting in $(B)_3$.

hypothesis that the transition can be updated to return to a consistent state. The transition is governed by the following three sets of equations: 3.1, 3.2, and 3.3, assuming the transition matrix is T, the parent marginal column vector is M_p with p values, and the child marginal column vector is M_c with c values.

$$M_c = T * M_p \tag{3.1}$$

$$\sum_{i=1}^{c} (col_j(T))_i = 1 \quad \forall \quad j = 1, \ldots, p \tag{3.2}$$

$$T_{i,j} \geq 0 \quad \forall \quad i = 1, \ldots, c \quad \text{and} \quad j = 1, \ldots, p \tag{3.3}$$

Due to equations 3.3 and 3.2, all values are automatically limited to the range $[0, 1]$, which

is necessary for transitions potentials, per the definition [17]. These equations do not fully constrain the transition potentials, thus requiring an optimization routine to choose a feasible solution which minimizes a cost function. The equality constraints (equations 3.1 and 3.2) can be rewritten into a vector equation of the form $Ax = b$, where A is a matrix and b and x are vectors. This equation can be optimized to solve for x. There are two important points in this optimization design:

- In most cases, there will be one redundant equation. The redundant equation is not known *a-priori* because it is dependent on the parent and child marginals. Therefore, all equations are included in the optimization, which does not degrade the solution.
- The inequality constraint, 3.3, is not included in the vector equations. Depending on the optimization routine and cost function chosen, this constraint may or may not be included.

The primary goal is to find a feasible solution which updates the network in a stable manner and is quickly computable. With that aim in mind, the cost function was chosen to be equation 3.4.

$$min \left(\sum_{i,j=0,0}^{i,j=m,n} \left(T^*_{i,j} - T_{i,j} \right)^2 \right) \tag{3.4}$$

This cost function minimizes the change from the previous transition matrix to satisfy the stable update goal. While other cost functions can be chosen, this cost function enables least squares optimization, which does not require iteration and satisfies the goal of fast computations. Note that other solution methods also meet the stated goals as discussed in Appendix C. It is clear, however, that the cost function is incompatible with the design vector. This issue is because the design vector, x, is based on Equations 3.1 through 3.3, but Equation 3.4 is the difference between the previous transition matrix and the new transition matrix. Thus, the design vector is modified to account for this difference, with the resulting offsets calculated from the previous transition matrix and added into the b vector.

Importantly, a least squares solution without constraint modifications does not guarantee all design variables are greater than zero, which is necessary to satisfy Equation 3.3. In practice, the cost function pushes the design variables towards positive semi-definite values and usually provides feasible solutions since the previous values were positive semi-definite. This shortcut was deemed necessary to improve the solution speed. However, when the least squares solution returns an infeasible solution, another method has to be chosen. The following two described methods were tested, and the second was chosen for its reliability.

Adhering to the goal stated previously, constrained quadratic optimization was avoided due to the uncertainty of the iteration time. Instead, the problem was modified for linear programming, which automatically guarantees positive semi-definite design variables and returns solutions quickly. Two modifications were made for linear programming. First, the design variables were changed to be the new transition potentials. Second, the cost function was changed to be the sum of the design variables with weights. Higher weights were placed on design variables that were previously close to zero to approximate the effect of least squares minimization. While this optimization resulted in a different solution than the least squares method, it provided feasible solutions with fast computations, which is necessary for real-time updates. However, in practice, the NumPy [100] implementation of this method was found to be unreliable at finding feasible solutions. Instead, a second method was developed to find feasible solutions.

The least squares solution always meets the constraints given by equations 3.1 and 3.2 since those are defined for the solution method, and it was previously shown that at least one solution always exists that meets all constraints. The task is then to adjust the solution as minimally as possible to meet the constraints defined in equation 3.3. This adjustment can be done through a series of mathematical operations defined below that (i) maintain adherence to the constraints in equations 3.1 and 3.2, (ii) find a result which meets the con-

straints in equation 3.3, and (iii) attempts to minimize deviations from the solution found via the least squares method. The goal is to modify all values that are negative to be non-negative. Due to the constraints given in equation 3.2, this automatically guarantees that any values greater than the value of one will also be reduced to less than or equal to the value of one. The procedure is as follows:

(1) Order the columns to adjust values. Do this by summing all values that are less than zero or greater than one in the column and sort from greatest to least. This order is useful because the maximum value this sum can attain is 1.0. This is because any value above 1.0 must have an equivalent set of values below zero to compensate. However, values below zero can be balanced by values in the range $(0.0, 1.0]$. Since this is true, when the sum is equal to 1.0, the scenario is the most highly constrained in which all negative values must be used to balance the value that is greater than 1.0. Note that any columns that already meet the constraint will sum to 0.0.

(2) Step through each column from step (1) above in order.

(3) For each column, order all values in the column that are less than 0.0 from maximum absolute value to minimum absolute value. While this order is not required, it provides a consistent solution method which makes debugging easier.

(4) For each value in order from step (3) above, redistribute excess mass per Algorithm 1.

As shown in Figure 3.7, this method can handle cases in which the child node of the transition being updated only has a single parent transition — case (1) in Figure 3.7. For multiple parent transitions — case (2) in Figure 3.7 — this update method is too simplistic since each parent transition would return the child marginals, with the combination of those marginals resulting in a different solution.

This effect means that the transition potentials for any parent transitions to the same child

Algorithm 1: Mass redistribution algorithm. This algorithm redistributes mass which does not adhere to the final constraint in Equation 3.3 in a way that seeks to minimize the difference from the minimum solution found through the least squares approach.

```
excess_mass = 0.0 - negative_mass_value
negative_mass_value = 0.0
if some_value > 1.0 in column then
    some_value -= excess_mass
else
    previous_positives = {}
    for all some_value in column do
        previous_positives[some_value] = some_value >= 0.0
        if some_value is not negative_mass_value then
            some_value -= excess_mass / (len(column) - 1)
        end
    end
    for any some_value in column do
        if (some_value is < 0.0) and (previous_positives[some_value] is True) then
            # Redistribute over remaining values per the above for loop
        end
    end
    # Repeat above until the mass is redistributed as close as possible to
      excess_mass / (len(column) - 1) since that is
    # the minimum squared changed from the previous value.
end
# Now that the column has been redistributed, redistribute the mass across the
  other columns to balance out equation 3.1. The goal is to equally distribute the
  mass across all columns, with compensation based on the parent marginals. This
  compensation ensures that changes do not drastically affect other columns,
  causing new issues.
for all some_column in columns do
    if some_column != adjusted_column then
        column_compensated_mass = adjusted_mass *
          parent_marginal[adjusted_column] / parent_marginal[new_column];
          some_column[adjusted_value] += column_compensated_mass
        # If some_column has not already been fixed, allow the value to go
          negative. Once it has been fixed, the excess mass must
        # be distributed across columns to ensure this value does not go negative
          while minimizing changes from applying the average
        # across all columns, which minimizes squared changes from the least
          squares solution.
    end
end
```

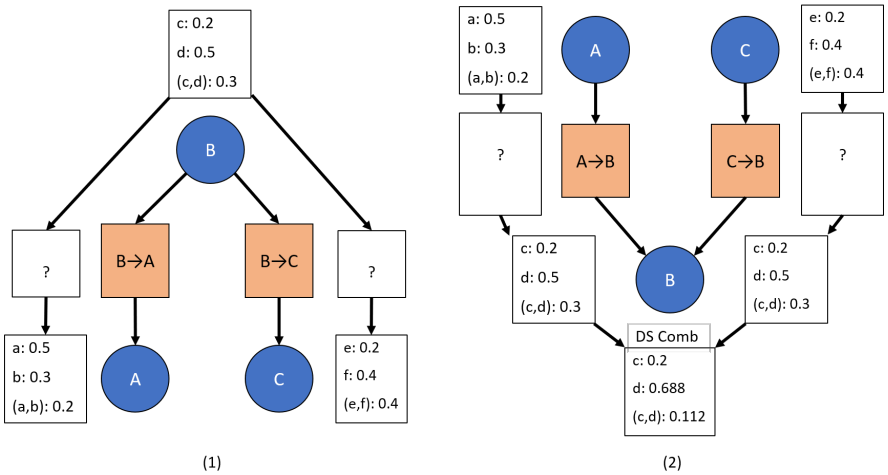

Figure 3.7: Case (1) shows a single parent node with transitions to two child nodes. Each unknown transition can be calculated using the update method described previously via either least squares minimization or linear programming minimization. Moreover, case (1) reduces to the simplest case of one parent and one child node if Node C and the associated transitions are removed. In contrast, case (2) cannot solely be solved via the described least squares or linear programming methods. The child marginal values are a result of a Dempster-Shafer combination algorithm, which must be part of the method for updating the unknown transition potentials. This case is handled in Section 3.3.3

node are linked and must be solved together. The naïve approach is to simultaneously solve for all transition potentials using equations 3.2 and 3.3. Equation 3.1 would be modified to equation 3.5.

$$M_c = comb\left(T_1 * M_{p1}, T_2 * M_{p2}, ..., T_n * M_{pw}\right) \tag{3.5}$$

In Equation 3.5, $comb$ is the DS combination algorithm chosen from the list of options discussed previously, and w is the number of parent transitions that must be simultaneously solved. The primary issue with this method is that DS combination methods can have highly non-linear effects depending on the evidence sets fed into the combination method. Further, because all transition potentials are being solved simultaneously, the optimization routine becomes significantly more complex to run, causing a large, non-deterministic increase in optimization time. To combat these issues with this solution method, two sim-

plifying assumptions are made:

- The simultaneous optimizer is used only to solve for the DS combination of marginals that results in the child marginals. This problem is significantly smaller than optimizing the full transition matrices with the DS combination algorithm included. Then, the solution methods for individual branches (the previously-described least squares and linear programming minimization methods) is used to find the correct transition potentials for each parent transition based on the specific marginals for that branch.

- With the assumptions that the DS network is initialized with completely unknown information and all transitions are learned as information is added to the network, then there is no *a-priori* information concerning the transition potentials. Thus, any solutions which meet the previously defined constraints of transition potentials are reasonable. Given this, the following simplifying assumption is made when updating the transition potentials matrix: all marginals combined via a DS algorithm to produce the desired child marginal are the same. Subsequently, this assumption can be relaxed for certain combination algorithms.

Given these two assumptions, an optimizer could be used on the reduced problem to find the marginals which combine to create the desired child marginal. An appropriate optimizer would be a trust-region interior points method or something similar. However, this optimization still suffers from non-deterministic run times and difficulty sectioning the optimization routine for real-time operating systems unless a specific optimization routine were written for this. Instead, a more reliable and faster method was developed for certain combination algorithms as detailed in section 3.3.3.

3.3.3 Multi-Parent Transition Updates

Several DS combination algorithms reduce to the same algorithm when combining identical evidence sets. In particular, the original DS rule [1], Murphy's rule [19], Zhang's

rule [18], and Evidential Combination Reasoning (ECR) [89] are the same for identical evidence sets and equal weights. This result is because (i) Murphy's rule and Zhang's rule reform the evidence masses per their rules, then combine the reformed evidence masses via Dempster's rule $(n-1)$ times, where n is the number of evidence sets; and (ii) ECR was specifically designed to reduce to Dempster's Rule when using equal weights [89]. Because Dempster's rule does not result in a normalized combined evidence set regardless of whether the input evidence sets are all normalized, Yager's rule [101], which is similar to Dempster's rule but assigns the unallocated mass to the universal set, does not fit the previous pattern, and the result developed in this section does not apply to Yager's rule [101]. While this result may seem fairly constrained since it only applies to four rules, it serves well for most decision networks, and in particular risk-analysis networks, since those four rules can cover the various cases to which decision networks are typically applied. The averages in Murphy's rule [19] apply when low beliefs of an event occurring are important. Conversely, Zhang's rule [18] applies when outliers need to be eliminated. The original DS rule [1] and ECR [89] apply when only the combined evidence is known at each update; the history of evidential inputs is not retained. Further, by combining Zhang's rule [18] with sufficient evidence history, the winning decision will be emphasized, which is one of the primary propositions of the Rayleigh methods [20]. Finally, the chief property of Yager's rule [101] — avoiding issues with Dempster's rule [79] — is also a property of Murphy's rule [19], Zhang's rule [18], and ECR [89].

Murphy's rule is intuitive for risk-analysis since it uses averages to reform the evidence masses [19], and averaging large datasets to obtain means and standard deviations is typical for risk analysis. Zhang's rule [18] reforms the evidence sets in a weighted fashion which helps to reduce or eliminate outliers which skew the data. While this rule can eliminate potentially important information in risk analysis (e.g. if failure rates are less than 0.001%, the failures may be classified as outliers and eliminated), it still has application in

risk-analysis. ECR provides a different capability which is essential for large, continuously expanding datasets. Zhang's Rule requires maintaining all the evidence and reforming at each update [18]. This method quickly becomes infeasible for large sets of evidence. Murphy's Rule, using averages, can be updated at each step. However, if the underlying average information is lost due to rebuilding the network, the ability to update the averages is also lost. In short, Murphy's Rule and Zhang's Rule work well for a framed problem (e.g. a sliding window of evidence over which the decision is being made), but break down when no frame is used, and the evidence becomes excessive or the underlying evidence is no longer available. Both Dempster's Rule and ECR add the new evidence into the previously combined data, thus bypassing both the framing and rebuilding issues. Because Dempster's rule produces counter-intuitive results [79] when evidence sets do not have equal weight, equal reliability, or visibility of the full powerset of inputs, the rule has less applicability to risk analysis. However, it is the basis of the other rules and the ,result that will be developed in this section applies to Dempster's rule [1] as well.

To develop the algorithm, first note that in Dempster's rule, input evidence masses can only apply to output masses that are a subset of the input mass. For example, (a, b, c) applies to all elements of the powerset, but (a, b) can only apply to a, b, and (a, b). Thus, for identical input evidence sets, the only contributor to the universal set (in the example — (a, b, c)) is the universal set, to the n power, where n is the number of times the identical evidence sets are combined. With no other dependencies, the input mass for the universal set is immediately solvable from Equation 3.6.

$$(universal \quad set)_{out} = (universal \quad set)_{in}^{n} \tag{3.6}$$

Continuing on with this trend, an example for two identical evidence set inputs and 3 options per input is shown in Table 3.4.

Table 3.4: Dempster-Shafer combination details for two identical input evidence sets with three options each. E1 and E2 are evidence sets one and two, respectively. Each matrix cell mass is assigned to the specified Destination row mass unless otherwise stated in the cell.

		E1					
		A	B	C	(A,B)	(A,C)	(B,C)	(A,B,C)
E2	A	A^2	0	0	$(A,B)\,A \to A$	$(A,C)\,A \to A$	0	$(A,B,C)\,A \to A$
	B	0	B^2	0	$(A,B)\,B \to B$	0	$(B,C)\,B \to B$	$(A,B,C)\,B \to B$
	C	0	0	C^2	0	$(A,C)\,C \to C$	$(B,C)\,C \to C$	$(A,B,C)\,C \to C$
	(A,B)	$(A,B)\,A$	$(A,B)\,B$	0	$(A,B)^2$	$(A,B)\,(A,C) \to A$	$(A,B)\,(B,C) \to B$	$(A,B,C)\,(A,B) \to (A,B)$
	(A,C)	$(A,C)\,A$	0	$(A,C)\,C$	$(A,C)\,(A,B) \to A$	$(A,C)^2$	$(A,C)\,(B,C) \to C$	$(A,B,C)\,(A,C) \to (A,C)$
	(B,C)	0	$(B,C)\,B$	$(B,C)\,C$	$(B,C)\,(A,B) \to B$	$(B,C)\,(A,C) \to C$	$(B,C)^2$	$(A,B,C)\,(B,C) \to (B,C)$
	(A,B,C)	$(A,B,C)\,A$	$(A,B,C)\,B$	$(A,B,C)\,C$	$(A,B,C)\,(A,B)$	$(A,B,C)\,(A,C)$	$(A,B,C)\,(B,C)$	$(A,B,C)^2$
	Destination	$\to A$	$\to B$	$\to C$	$\to (A,B)$	$\to (A,C)$	$\to (B,C)$	$\to (A,B,C)$

When converted to equation form, the equations for each of the combined evidence masses is a polynomial of order n, where n is equal to the number of identical evidence sets entered

into the combination algorithm. Further, each combined evidence mass is only dependent on inputs of the same evidence mass and on evidence mass inputs closer to the universal set. For example, in Table 3.4, the resulting mass a_{out} is only dependent on a_{in}, $(a,b)_{in}$, $(a,c)_{in}$, and $(a,b,c)_{in}$. This set of polynomial equations can be solved individually if done in the correct order, which results in an easy algorithm to run as well as one that can be paused mid-update for real-time operating systems.

Expanding this example further, the general case is shown in Equations 3.7 through 3.8, where o means out, i means in, n is the number of input evidence sets, $universal \quad set - p$ is the subsets of number of elements of the universal set minus p elements, and $g_{...}$ are the multipliers for each polynomial term and follow the pattern defined in Algorithm 2. One noteworthy point is that the simplest cases, such as polynomials of order two, are easy to solve directly. At higher orders, root finding methods are easy to use. However, since each result is used to compute the results of the next combined evidence masses, errors compound. Thus, for larger power sets and higher order polynomials, root finding errors will eventually limit the utility of this solution method unless more precise computations are executed.

$$(universal \quad set)_o = (universal \quad set)_i^n \tag{3.7}$$

$$\begin{aligned} &\forall m_o \in (universal \quad set - p)_o \quad m_o = \\ &m_i^p + g_{p-1}\left(\sum \forall q_i \in (universal \quad set - r), r = [0, p-1] \quad \& \quad q_i \cap m_i = m_i\right)_i^1 m_i^{p-1} \\ &+ \cdots + g_1\left(\sum \forall q_i \in (universal \quad set - r), r = [0, p-1] \quad \& \quad q_i \cap m_i = m_i\right)_i^{p-1} m_i^1 \\ &+ \left(\sum q_i \in (universal \quad set - r), r = [0, p-1] \quad \& \quad q_{i_1} \cap q_{i_2} \cap \cdots \cap q_{i_n} = m_i\right) \end{aligned} \tag{3.8}$$

Algorithm 2: Multiplier calculation algorithm. The polynomial multipliers are based on the equations detailed above. This algorithm provides a simple method for calculating these multipliers in code.

```
level = 2
multipliers = [2]
for level in range(2, p) do
    multipliers_old = multipliers
    multipliers[0] = multipliers_old[0] + 1
    for index in range(1, len(multipliers)) do
        multipliers[index] = multipliers_old[index] + multipliers_old[index - 1]
    end
    multipliers.append(multipliers_old[len(multipliers_old)] + 1)
end
```

Recalling that all focal points m_i must be positive semi-definite, the equation above presents a bound on the capabilities of this solution. Specifically, the zero-eth power term in equation 3.8 must be less than or equal to the child marginal, m_o. If this is not the case, then the equation returns a negative solution. Assuming that the child marginal is a combination of identical parent marginals — as in the case of Murphy's Rule or Zhang's Rule — of at least the number of parents, then a valid solution will always be found via this method. In practice, this limitation is not an issue. This limitation simply means that the child node must have enough evidence sets combined to be at least the number of parent nodes. In effect, the solution method cannot guarantee a valid solution until the node is sufficiently initialized. In many cases before the initialization is complete, a valid solution is available even though it is not guaranteed. Within the initialization, the case in which all parents but one have the total mass in the complete set is trivial, since the remaining parent can be set equal to the child. Other situations only arise when using Dempster's Rule or ECR. Currently, a solution is not available to calculate the parent marginals without using a high dimensionality optimization, which is a slow process. Instead, the focus is placed on restricting the inputs to ensure that the child marginals have an available solution for the parents. This restriction is accomplished by analyzing the polynomial solution discussed in Equations 3.7 through 3.8. The restriction in Equation 3.9 and 3.10 allows Dempster's

Rule and ECR to retain the ability for the reverse calculation for multiple parents, assuming that the previous assumption holds of at least as many inputs as parents. Validation of this restriction is shown in Appendix A.

$$\forall m_{a_{in}} \quad except \quad universal \quad set, m_{in} \leq m_{b_{in}} s.t. m_{b_{in}} \subset m_{a_{in}} \tag{3.9}$$

$$m_{(universal \quad set)_{in}} \leq \sum m_{(a,\ldots)_{p-1}} \quad where \quad p = len\,(universal \quad set) \quad and \quad a \subset \Theta \tag{3.10}$$

By combining the equations and algorithms developed in this section with those in Section 3.3.2, updates to the nodes in the DS network can be used to learn the transition potentials between all nodes in the network. While these updates can be performed when a single node is observed and other nodes are updated based on inference, more reliable updates are performed when more than one node is observed simultaneously. However, these updates present a conundrum since each node update will propagate throughout the entire network, thus providing the effect of several updates simultaneously. The solution to this issue lies in proper weighting of the updates, which is discussed in Section 3.5.

3.4 Episodic Learning

As shown in Section 3.6, the learning methods detailed in Section 3.3 enable the DS network to understand the relationship between nodes based on evidential inputs at each node. This method is sufficient for some applications: when the windowed mass distribution at each node represents the entire mass distribution of interest for understanding the network relationships. Recall that the window can be defined as long as appropriate for the application. With reasonable weighting (see Section 3.5), the window can encompass the entirety

of the evidence history.

However, this view is limited. In many cases, the entire mass distribution is not visible in the current window, and including the entirety of the evidence history does not capture the nuances of the relationships. For example, consider a traffic light scenario in which the network is evaluating the relationship between the time until the light turns green and the state of the cross-traffic light (see Figure 4.1 for the scenario). The relationship would be roughly expected as shown in Table 3.5.

Table 3.5: Expected relationships between cross-traffic light and time until green. Given knowledge of the cross-traffic light at an intersection, this table details the expectations of the time until the light changes from red to green for the evaluator to continue through the intersection. Note that any ambiguous sets are removed for ease of description. Those sets can be interpolated from the relationships shown in this table.

	Green	Yellow	Red
Long	0.9	0.1	0.0
Medium	0.1	0.8	0.1
Short	0.0	0.1	0.9

The issue with learning these relationships is that the entire distribution is not present at a given time. When the cross-traffic light is green, estimates of time until the same-side light changes from red to green are likely only long and medium. Likewise, when the cross-traffic light changes to red, estimates of time until the same-side light changes from red to green are likely only short and medium. Using the previously developed learning methodology in Section 3.3, the natural response would be to retain all evidence via Murphy's rule using a weighting scheme. However, this method will be biased by the amount of time spent in each situation. For example, assuming the light is being evaluated from a long distance away approaching the intersection, one can safely assume that the evidence gathered usually reports the cross-traffic light as green and the time until the same-side light changes from red to green as long. In this case, the learned relationship will be biased towards a long time until green regardless of the current state of the cross-traffic light because the

majority of evidence is during a period in which it takes a long time for the light to change to green (i.e. other relationships are masked by the amount of evidence showing a long time to green, even though much of this evidence is just repeating known information). In order to combat this issue, an episodic learning method was developed, which required two additional mathematical rules.

3.4.1 Change-Weighted Least Squares

Recall from Section 3.4 that the goal is to capture in the transition the nuances of the relationship between the mass distributions of the parent and child nodes. Note that while this discussion is applied to a single parent and child, it can be generalized to multiple parent or child nodes using the techniques developed in Section 3.3. The question then becomes how to mathematically capture the concept of only updating the weights in the transition that apply to the current episode. Suppose the baseline distributions for the parent and child nodes are known. With no loss of generality, we will take those baseline distributions as the unknown distribution: $p = 0 \forall p \in M \wedge p \neq universal \quad set$ and $p = 1 \iff p = universal \quad set$, where M is the marginal mass vector of the parent or child. Then, modify Equation 3.4 to include weighting relative to the change from the baseline distribution, as shown in Equation 3.11, where c is an arbitrary control term for determining the effect of the weighting. Note that Equation 3.1 is modified with the weights defined in Equation 3.11 to balance the equations.

$$min \left(\sum_{i,j=0,0}^{i,j=m,n} \left(\left(T^*_{i,j} - T_{i,j}\right) * \left(1 + \frac{c}{\left(M^*_{p_i} - M_{p_i}\right)^2 * \left(M^*_{c_j} - M_{c_j}\right)^2} \right) \right)^2 \right) \tag{3.11}$$

Observe three points concerning Equation 3.11:

- If the control constant c is set to 0, then the additional weight disappears, and the

equations reduce to the previous set.

- The weight is inversely proportional to the squares of both the differences in applicable parent marginals and the applicable child marginals. If the child or parent marginal changes minimally, then the design vector is weighted such that the applicable transition value should be modified minimally from its current value.

- If either the parent or child marginal change is zero, then the inverse weighting is undefined — divide by zero scenario. Thus, this weighting must be protected by a maximum weight. In practice, any weight on the order of 1,000 or above does not have much effect since there is a limit to the flexibility of the solution given the rest of the constraints.

3.4.2 Zero Marginal Value

Recall from Equations 3.1 to 3.3 that a zero marginal value in the child marginal vector means that all non-zero values in the parent marginal vector must be multiplied by a transition value of zero for the equations to balance. Assuming marginal probabilities are reset to a baseline between episodes in order to capture the effects of each episode, it is reasonable to assume that marginal values will often be zero even if those values were observed in prior episodes. In practice, this can cause prior episodic information to be lost. To avoid this loss of information, any marginal value which was observed in a prior episode is prevented from returning to zero and is instead reset to a small value in the range of $[0, 1]$, such as 0.01. Due to the weighting described in Section 3.4.1, such a small value will result in a high weight, preventing the associated transition values from changing significantly.

3.4.3 Episodic Learning Implementation

Given the additional mathematical rules described above, episodic learning is implemented as shown in Algorithm 3. Determining episodes can be accomplished either through expert

information or automatically by determining when marginals are statistically different than previous episodes.

Algorithm 3: Episodic learning algorithm for a Dempster-Shafer network. This algorithm captures episodes in order to only update the portion of the transition potential matrix to which the episode applies.

```
reset_nodes_and_transitions_to_unknown()
while training is True do
    episode_name = counter
    counter += 1
    while change_in_marginals() ≈ 0.0 do
        step_training()
    end
    save_episode()
    combine_marginals_with_similar_episodes()
    calculate_transitions()
    reset_nodes()
end
```

Evaluation of this method is tested in Section 3.6.6 and detailed in Chapter 4, which explores the finer points of a scenario that requires episodic learning. One final point with episodic learning is that it can be used to determine whether current evidence is shifting away from historical evidence, suggesting that historical evidence is less reliable. Current evidence is added to the matching episode, if one exists. Since the episodes are matched by parent node, changes in the distribution of the child nodes can suggest a shift from historical evidence, which can be used to determine whether an alternate weighting scheme, such as detailed in Section 3.5, is appropriate to emphasize current evidence over historical evidence.

3.5 Evidence Weight

Traditional DS evidence combination assumes equal weights and reliabilities between all evidence sets and contributors [1]. Likewise, most modifications to the DS combination rule make the same assumption. An exception is the Evidential Reasoning rule which han-

dles both reliability and weight of evidence sets [89]. While a single DS node handles the assumption of equal weights well, the network requires the ability to weight evidence. To understand this requirement, observe the update to the network in Figure 3.8. Assuming that all inputs propagate through the network, each input equally affects all nodes, although the data is inserted as observed at a single node and inferred at others. Further, although all observations may be a result of a single update, for example, the results of a one-hour test flight. Each input would result in an update at all nodes, introducing the equivalent of y hours of test flight data, where y is the number of observations for the network from the one-hour test flight. While more node observations for the one-hour test flight likely results in more accurate information, the data arguably should not have the same effect as multiple hour test flights with observations for each of those hours. Note that this discrepancy was previously handled by a discounting scheme to handle the reduced reliability of inferred evidence versus directly observed evidence [95]. This scheme is similar to Shafer discounting [3]. While useful, no rules were developed to define the discounting factor [95]. Further, discounting has additional effects on evidence combination [79], which must be handled. This section develops a new rule that explicitly defines the weighting factor based on the network input evidence, and the weighting factor can be calculated explicitly from the network for each update. Further, weighting methods (discussed in Appendix B) are explicitly developed for each combination rule used, avoiding known issues with Shafer discounting [89].

To resolve this discrepancy between the weight of evidence and the time of observation relative to pre-existing data, a weighting scheme is proposed, as demonstrated in Figure 3.8, part (2). At observed nodes, only the observation is entered with full weight. No data calculated through transitions is entered at observed nodes. For all other nodes, the weight of data calculated through transitions is one divided by the number of branches with data entering that node. The resulting weight for a single update is equal to one for all nodes.

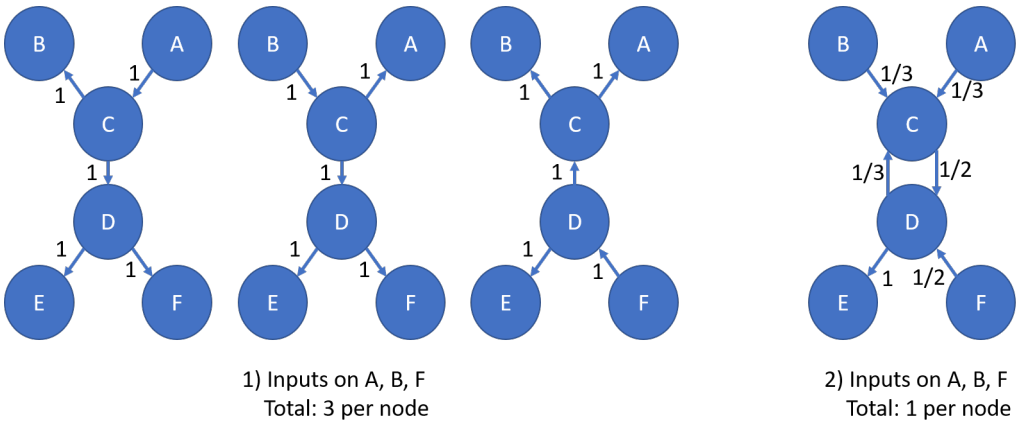

Figure 3.8: The results of weighting inputs for a Dempster-Shafer network. Given an update on nodes A, B, and F, a no-weight update results in 3 times the amount of experience applied at each node, as shown in part (1) of the figure. With weighting, only the applicable experience is applied at each node, as shown in part (2) of the figure. The direction of the arrows shows the transition of the update experience between nodes in the network. The network is using the standard representation, where moving upwards on the diagram between nodes represents inference.

Use of the weighting scheme has two implications. First, either only the Evidential Reasoning combination method can be used, or other methods must be modified to handle weighting. Second, given that weighting is necessary, further weighting methods can be used to improve update stability as discussed subsequently. Modifications for weights are easy to add to Murphy's Rule, Zhang's combination method, and the Rayleigh combination methods. Details on these modifications can be found in Appendix B.

Table 3.6 shows why additional weighting schemes are necessary since a single event, regardless of the length of operation time it represents, can significantly impact the per-hour risk if weights are not adjusted for the evidence sets. Since the combination scheme impacts the results as well, the Rayleigh and Murphy results are still significantly different regardless of weighting method. Consequently, risk analysis results thrash between extremes as differing events are entered into the network. Instead, a weighting scheme based on completed flight hours is defined. For each network node, the accumulated hours of experience represented by the combined evidence is stored. New evidence sets are relatively

weighted as $\frac{operation\ \ hours}{total\ \ operation\ \ hours}$. This concept is similar to how new values are added into an average: by multiplying the average by the total experience it represents and adding the new values before dividing by the new total experience. In both cases, the weighting scheme prevents new evidence from over-influencing previous experience.

Table 3.6: Example of why additional weighting schemes are necessary. "Evidence 1" has significantly higher weight than "Evidence 2". Without an included weighing scheme, the resulting combined data in "Rayleigh" and "Murphy" shifts significantly away from "Evidence 1" even though "Evidence 2" should arguably have a smaller impact on the final result. The resulting combined data in "Weighted Rayleigh" and "Weighted Murphy" is much closer to "Evidence 1" than "Evidence 2", which is expected given the relative weights of the two evidence inputs.

Data Set	Experience	A	B	C	(A,B)	(A,C)	(B,C)	(A,B,C)
Evidence 1	30.0	0.5	0.1	0.2	0.0	0.05	0.05	0.1
Evidence 2	0.5	0.1	0.05	0.4	0.05	0.2	0.1	0.1
Rayleigh	n/a	0.47	6.1e-4	0.24	2.6e-4	0.28	1.2e-3	2.2e-4
Weighted Rayleigh	30.5	0.91	3.8e-5	2.2e-2	1.4e-5	6.9e-2	1.6e-4	2.2e-3
Murphy	n/a	0.30	0.03	0.30	0.025	0.13	0.075	0.10
Weighted Murphy	30.5	0.49	0.011	0.20	8.2e-4	5.2e-2	5.1e-2	0.10

Two further available weighting modifications were evaluated and were found to have different uses.

- Recency: new evidence can be weighted higher than the relative weight to the total experience for risk analysis in which more recent information is considered more reliable. In particular, this weighting scheme is advantageous if historical data potentially introduces a bias or is deemed less reliable. One way to capture this historical versus current evidence weight is to retain a maximum number of evidence sets, and define the final evidence in that set as the total "historical" evidence. That total "historical" evidence can be weighted as the total weight of the historical evidence multiplied by some reduction factor to account for the lower reliability of the evidence.

- Inference: an additional weight reduction can be applied across all transition inferences assuming that inferred evidence is less reliable than directly observed evidence.

This method has been used previously [95], although the weights are explicitly now defined for each Evidential Reasoning combination method.

3.6 Dempster-Shafer Network Results

A DS network was developed in Python to test the algorithms developed in Section 3.3. The code is available on GitHub at "https://github.com/chacalnoir/DSImplementation" and "https://github.com/chacalnoir/DSNetwork". Testing covered the following areas:

- Single node updates with and without learning transitions and with and without weighting. These tests include modifying the transition potentials, learning from a completely unknown starting point, and multi-level learning (i.e. with unobserved nodes between nodes with observations). See Section 3.6.3.
- Multiple parent updates with weighting and learning transitions. See Section 3.6.4.
- A complex network to better understand the speed and effects of evidence propagation in a more realistic DS network. See Section 3.6.5.

3.6.1 Testing Methods

For each test, evidence sets were randomly generated. The same evidence sets and order of input were used for all combination algorithms at each test to provide consistency. Each test consisted of 30 updates of 3 simultaneous evidence sets per update. The nodes to which the evidence sets were injected were randomly generated as well. The number 30 was chosen to allow randomly generated sets to provide acceptable analysis metrics within a reasonable evaluation time. Furthermore, 30 tests were conducted per test case. For each evidence set, the complete set (unknown information) mass was set to 0.0, which ensured that the learning capabilities of the networks were properly evaluated against the metrics discussed in Section 3.6.2. For cases without learning, approximately 80% of transition

potential values were randomly set, since, without learning, the majority of transitions need to be set for the network to make sense. For learning cases, all values were defaulted to all belief mass assigned to the complete set (i.e. completely unknown starting points). The primary limitation of these tests is that the evidence sets are randomly generated. DS theory is designed to combine evidence sets for a given situation to arrive at a conclusion. As such, some consistency between the evidence sets is assumed. In fact, complete conflict, as discussed in Section 3.2, produces incalculable results for Dempster's Rule and the ECR rule without weighting. Due to round-off error and pseudo-random selection of evidence sets, it was assumed that complete conflict would not occur. However, since combination algorithms handle conflict differently, results are generally only valid when analyzed in comparison between tests using the same combination algorithm.

3.6.2 Metrics

The DS network developed in Chapter 3 was evaluated for the following goals: propagation time for implementation in real-time on an embedded system, learning for the ability to start from a state with completely unknown information, and explainability for the decisions to be accepted by organizations that use risk analysis.

- Propagation time per update per node: comparison between the baseline implementation and specific novel additions.
- Failures: how many of the tests fail, and the explanation for each failure.
- Consistency: the L2 norm of the error between the parent marginals multiplied by the transition potentials and the child marginals. Note that for a multi-parent node, the results of the parent marginals multiplied by the transition potentials are then combined via the appropriate DS combination algorithm.
- Unknown fraction: the fraction of node marginals and transition potentials that remain in the complete set (unknown information) at the end of the update set, given

that the network started with all masses in the complete (unknown) set.

- Weight per node: the total weight/experience attributed to each node at the end of every update.

The propagation time metric provides insight into the ability to implement in real-time on an embedded system. Likewise, the unknown fraction metric provides insight into the learning capability of the network and the ability to start from an unknown state. Less intuitive are the consistency and weight per node metrics, which provide insight into the explainability of the network. Weight per node maps the "experience" per node after the updates. For example, if each update represents one flight test hour, and 30 updates are made, each node should show approximately 30 hours of experience. Significant increases from that suggest that information is used more than once, which calls into question the validity of any conclusions.

The final metric, consistency, evaluates the mathematical explainability of the network at the end of all updates. Given a simple network where a parent node "A" connects to a child node "B", an inconsistent situation is one in which, after evidence propagation through the network, the marginals of "A" multiplied by the transition potentials do not equal the marginals of "B". From an explainability perspective, if only the final network state is viewed by the organization that needs to accept the decision, it is unclear from where the decision came and whether the network operated correctly. In short, the result is unexplainable. Since the original DS network implementation overwrite data at each node after propagation, an "overwrite" algorithm is included as the baseline for comparison.

3.6.3 Network Updates with Single Parent Only

This test analyzes the performance of the developed algorithms when applied to networks that only have single parents per child. The test network is shown in Figure 3.9. Each

DS combination algorithm was analyzed using this test network including an "overwrite" algorithm that does not combine evidence but rather uses the newest evidence set to provide a baseline. This test case analyzed the following options: learning transitions, single versus multi-node updates, and weighting inputs.

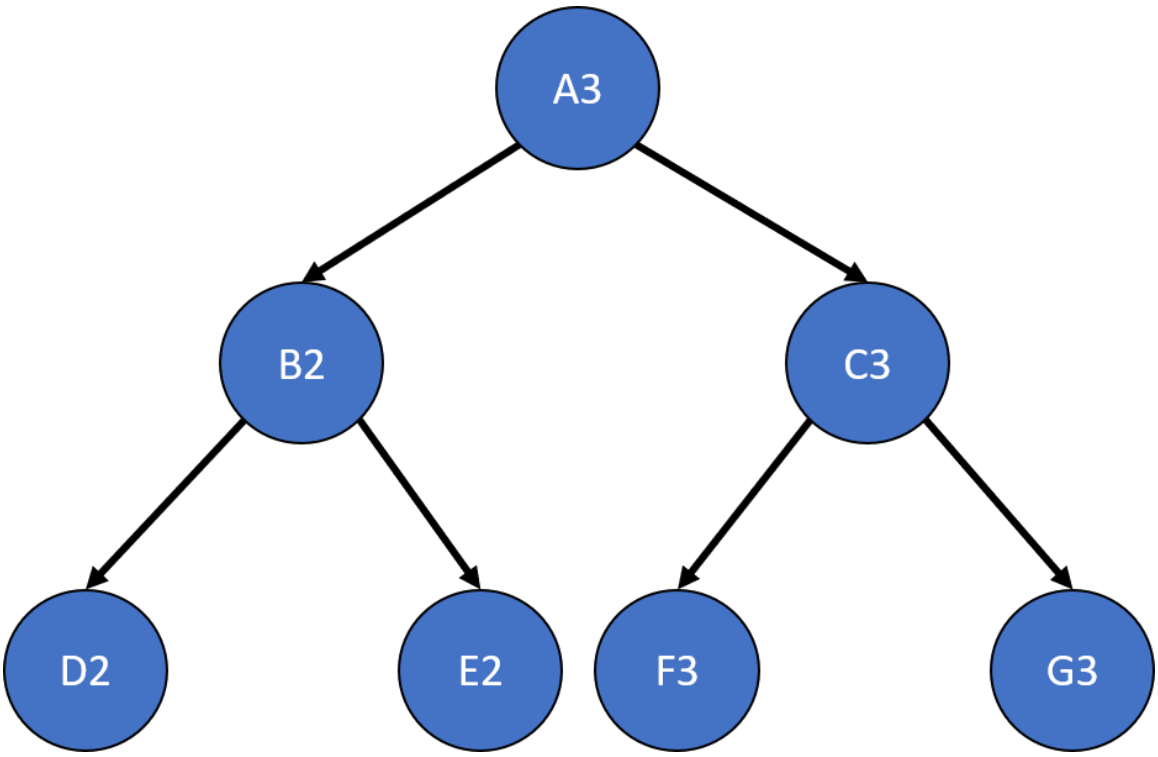

Figure 3.9: Test network used to analyze the performance of the novel Dempster-Shafer network algorithms. Only includes single parents for each node. The number after the node name shows the number of θs for the node. Two and three θ nodes were used since these are the more common cases for nodes in a DS network.

The results of these tests are shown in Figures 3.10-3.13. Figure 3.10, showing the time test results, emphasizes that the weighting methodology helps to overcome the additional burden of the learning calculations. In all combination algorithms, weighting improves the update time to only $[1, 2]$ orders of magnitude greater than the no-learning overwrite case, which is the baseline (10^{-3} and 10^{-4} versus 10^{-5}). Compared with the unweighted cases, which are always in the mid to high 10^{-3} order of magnitude, this is a significant improvement, thus potentially enabling single-parent network implementation in a real-time embedded system. In the second result, Figure 3.11, the learning method returns

an obvious improvement over all non-learning methods, resulting in effectively perfect consistency with round-off error and a high degree of explainability. In the third result, Figure 3.12, the three learning cases clearly improve upon the unknown knowledge in the network, even given that the no-learning cases started with a high degree of known transition potentials and the learning cases started with fully known information. This result is dependent upon the number of evidence sets injected, but the potential to reduce unknown information is shown clearly through this test. The final result, Figure 3.13, is perhaps the most straight-forward. The weighting method clearly shows approximately 30 units of experience/weight for 30 updates of 1 unit of experience, as expected. All unweighted methods show 3X units of experience per node, which suggests information was used 3 times for each update. This result brings into question the validity of results obtained from networks using unweighted updates since reusing the same information in DS logic changes the results.

3.6.4 Multi-Parent Learning Results

Based on the results of the single parent tests, the multi-parent tests are set up to only test learning with weighted simultaneous updates. The additional complexity that these tests add is the need to calculate multiple simultaneous parent marginals through the use of an optimizer or the root finding method detailed in Section 3.3.3. The test network used is shown in Figure 3.14. The results are shown in Figures 3.15 through 3.19. In most cases, no significant deviation is observed between the optimizer and root finder, especially for the combination methods most likely to be used. The exception to this statement is the run time. The primary reason for developing the root finder is that it is significantly faster than the optimizer (at least one order of magnitude per node), and it is deterministic, especially for two and three option nodes requiring quadratic or cubic solutions, which are closed form. For a real-time implementation on an embedded system, the root finder can be reliably scheduled and interrupted as necessary while operating in a way that is difficult

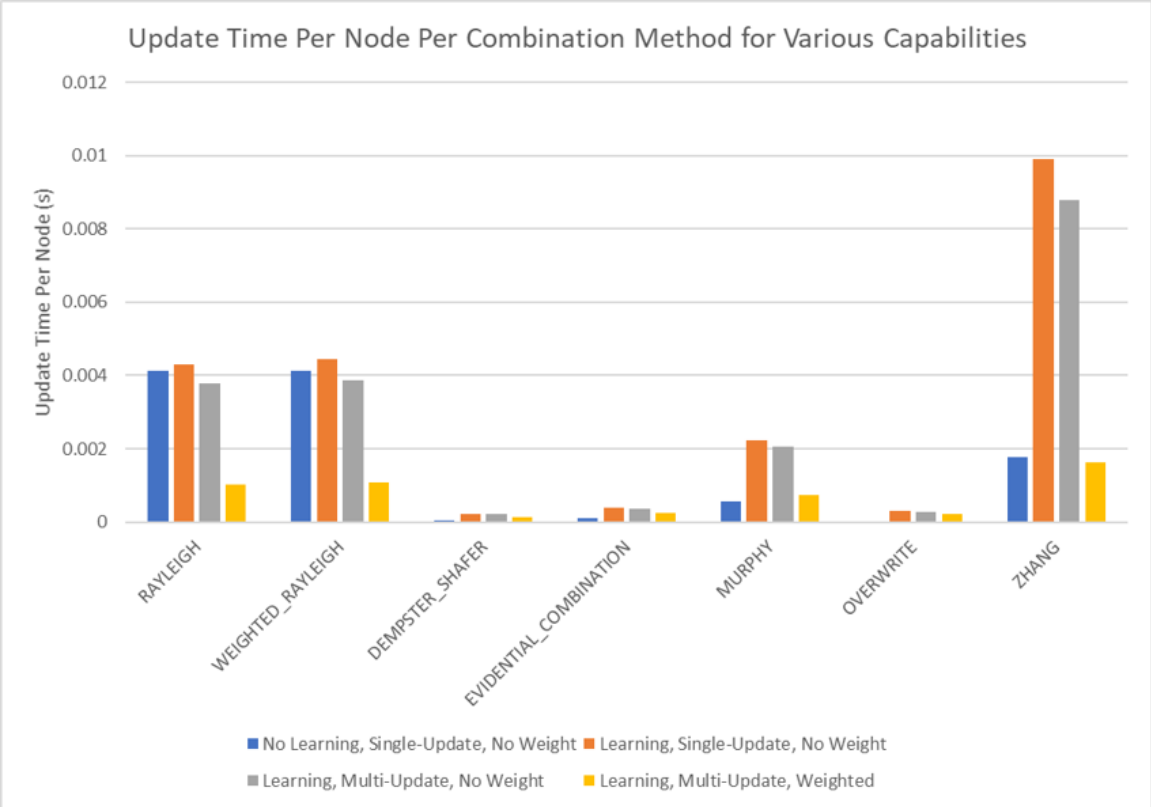

Figure 3.10: Test results for run time with single parent network. Within each combination algorithm group, the no-learning method is faster that un-weighted learning methods, which shows the increased burden of the learning calculations. However, both Rayleigh and Weighted Rayleigh methods show that the no-learning time is at the same order of magnitude as the un-weighted learning cases, suggesting that the combination algorithm is the driving factor for the update time. Since Dempster-Shafer, ECR, and Overwrite do not retain explicit history, their update times are significantly faster. Note further that single versus multi-update methods do not change the update times significantly when both are unweighted. Finally, the weighted methods are approximately equivalent to the no-learning update times, and, in some cases, are an improvement.

for generic optimizers.

3.6.5 Complex Network Results

The final tests are performed on the complex network shown in Figure 3.20. Within these tests, the comparison is made between unweighted single updates and weighted multiple simultaneous updates. In this case, it is also instructive to compare these results with the

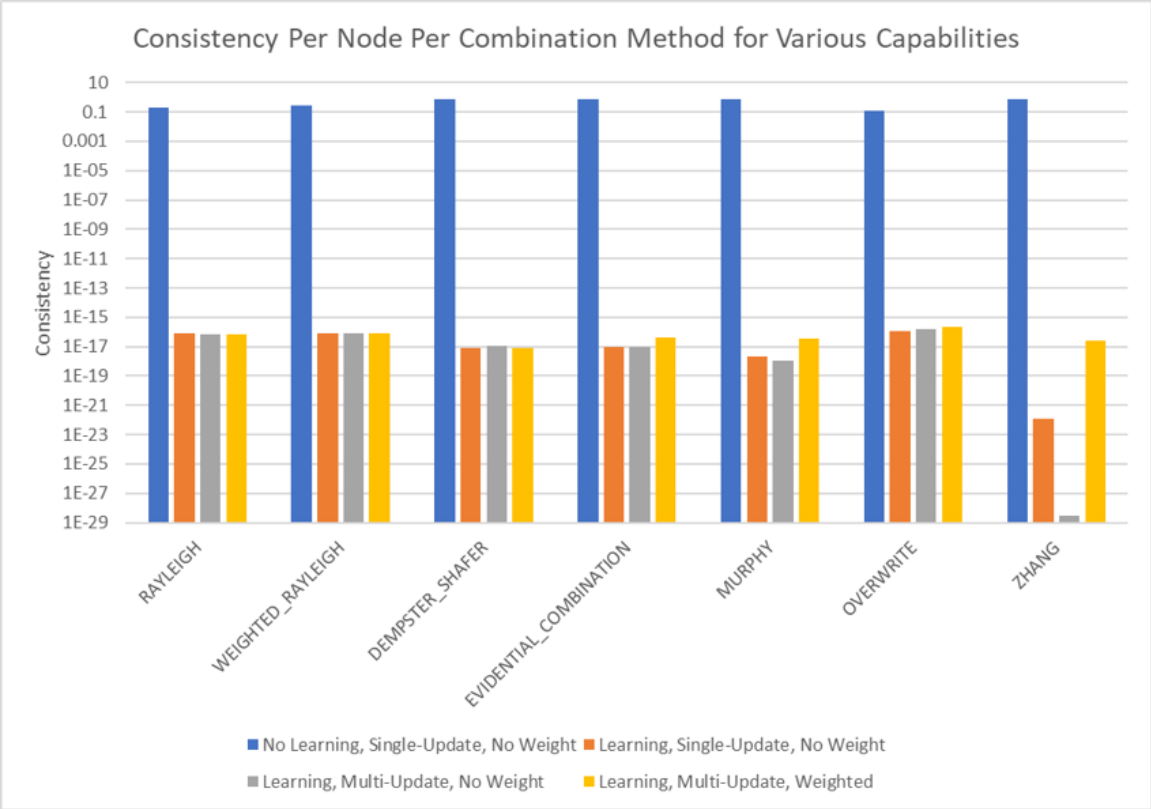

Figure 3.11: Test results for consistency with single parent network. The y-axis uses a logarithmic scale. Without learning, all per-node consistency is low, with the best case being the overwrite case. The reason that the overwrite case does not give an approximate zero consistency result is that transition potentials are not reversible in the consistency test. Propagating up the transition does not mean that a consistency check down the transition will return nearly perfect consistency. However, in all learning cases, the consistency checks return an effectively zero result, equating to perfect consistency with round-off error.

previous tests. Overall, the complex test case falls between the single parent and multi-parent test cases, as expected, given that it is a combination of the two. The complex test case is more similar to the multi-parent test case, suggesting that the multi-parent root finding method properties tend to dominate the metrics in these tests. This test has implications for scaling to more complex problems. First, computation times per update per node are fairly consistent, suggesting that overall computation time scales linearly with the number of nodes. Trials for nodes with more θs per node were not conducted since structure was specifically used to reduce the number of θs per node, and since DS computations

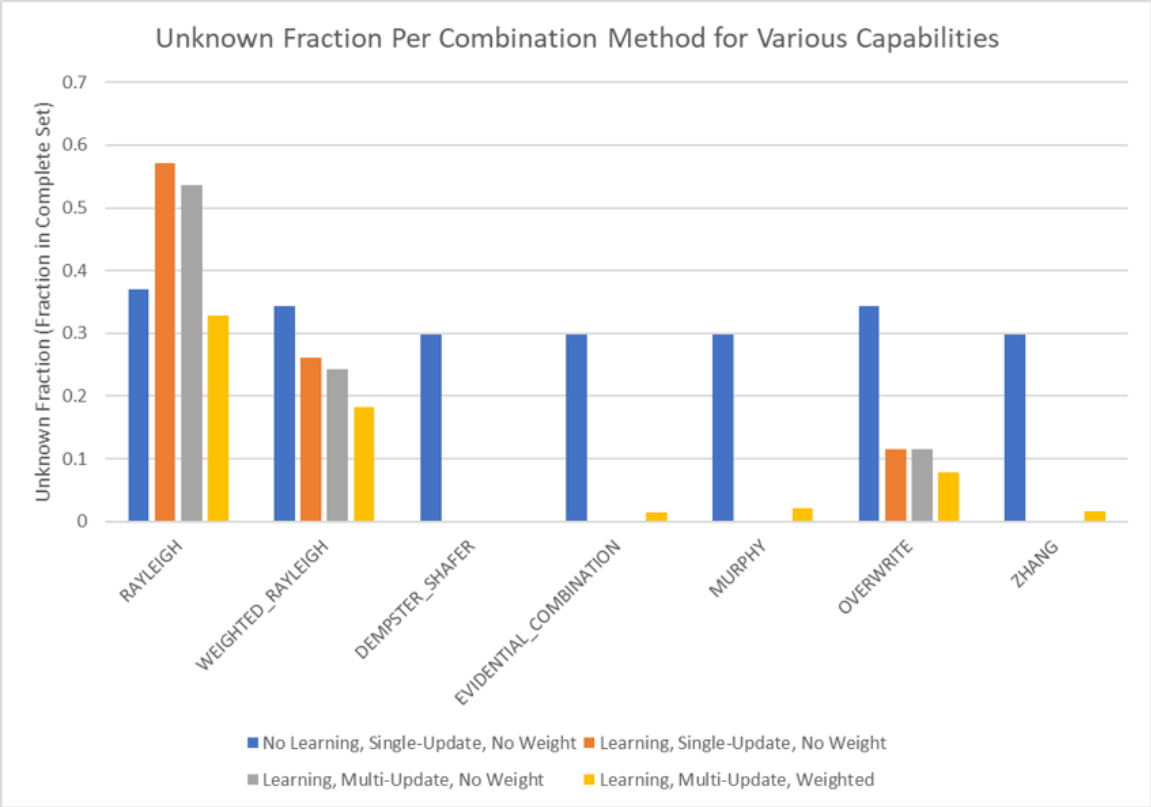

Figure 3.12: Test results for learning with a single parent network. Results vary depending on the combination algorithm used. Except for Rayleigh and Weighted Rayleigh methods, and to a lesser extent the overwrite method, the learning method resulted in significant reduction of unknown information over the baseline case of no-learning with preset transition potentials. Deviations in the combination algorithms can be explained through handling of conflict. High conflict in the Rayleigh and Weighted Rayleigh algorithms means lower assignment of mass to the new focal elements, resulting in higher mass retained in the unknown/complete set. This is a result of the randomized evidence set testing methodology and should not be construed as evidence for or against the combination methods. Finally, since the overwrite method does not retain previous evidence, propagated evidence through unknown transitions will tend to have higher impact, retaining unknown information. Since this method was included primarily as a baseline for the no-learning case, learning with this method is not expected to be used.

are already known to scale poorly with the number of θs due to the use of the powerset. Consequently, scaling to more complex problems in terms of computation time is a balance through using a network to maintain a low number of θs per node while not losing important interactions between θs.

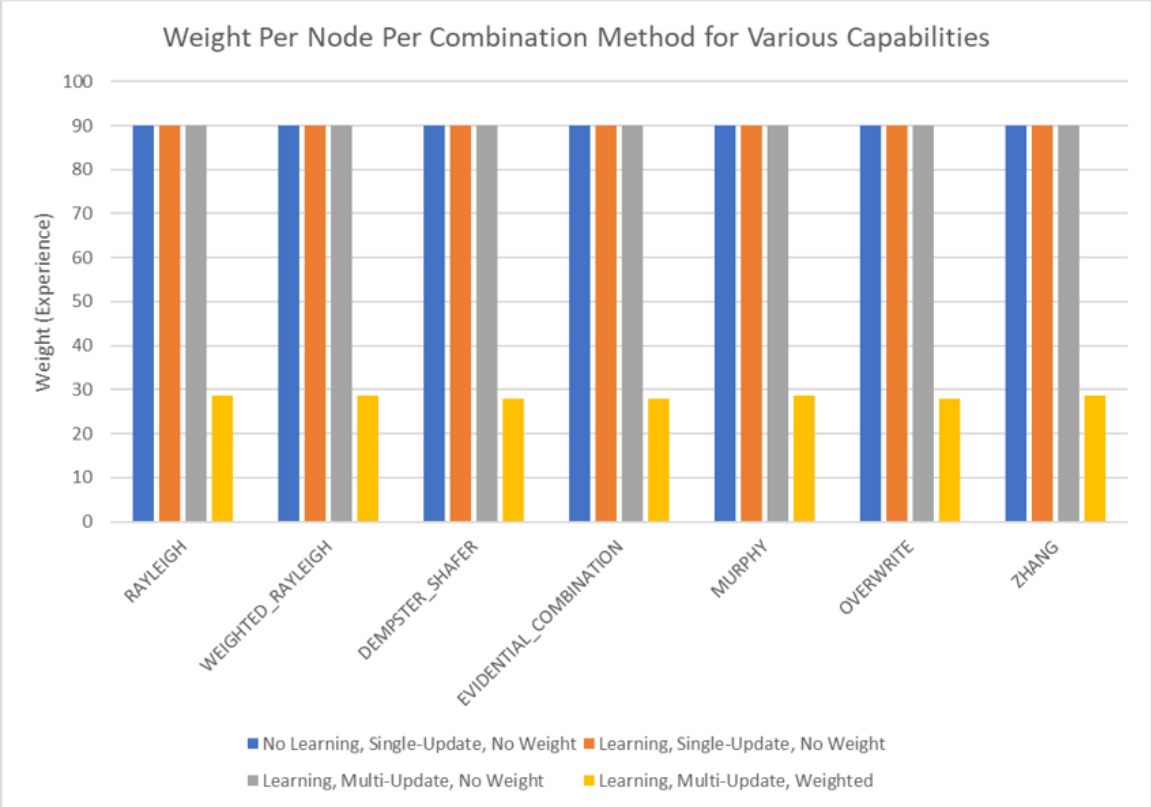

Figure 3.13: Test results for weighted versus unweighted methods in a single parent network. This test shows a clear demarcation between unweighted methods, with 90 units of experience/weight per node for 30 updates of 1 unit of experience each, and the weighted method, with approximately 30 units of experience/weight per node for 30 updates of 1 unit of experience each. The unweighted method clearly suggest that data is reused. While this is actually not the case (each propagated evidence set is from a different observation), this result is significantly less explainable than the weighted method, calling into question the ability for the network results to be accepted in decision-making scenarios.

3.6.6 Episodic Learning

Tests for episodic learning were conducted to determine whether this learning method better captured expected relationships than the least squared method without episodes. A two-node network was constructed with "Node 1" as the parent and "Node 2" as the child. Both nodes and the transition potentials were initialized to unknown and vacuous, respectively. Two evidence sets were then entered as defined in Table 3.7. The baseline test case without episodic learning injected the evidence sets to their respective nodes and update the

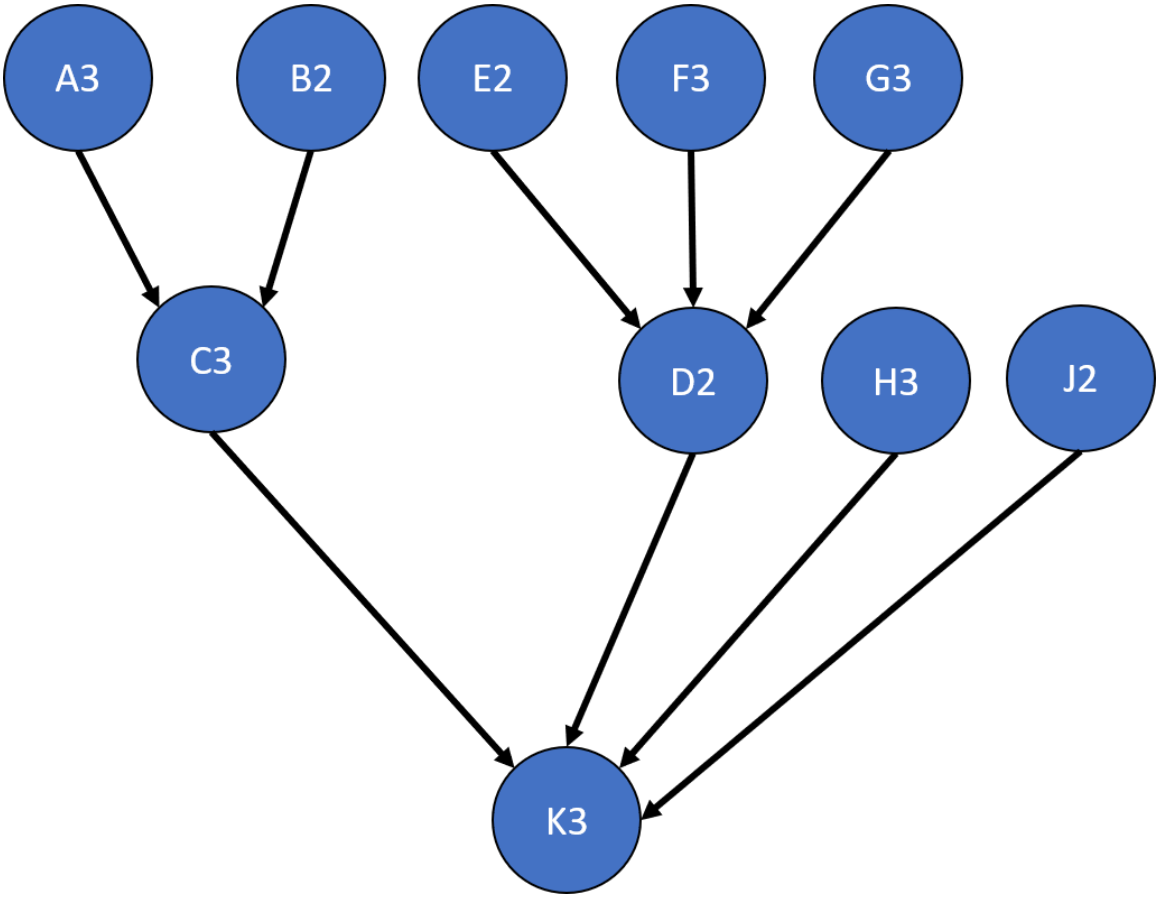

Figure 3.14: Test network used to analyze the performance of the novel Dempster-Shafer network algorithms. This network only includes multiple nodes per parent. The number after the node name shows the number of θs for the node. Two and three θs nodes were used since these are the more common cases for nodes in a DS network.

transition potentials matrix after each update. The results are shown in Table 3.8. The network was then reset, and the episodic test case was run, with the two evidence sets injected to their respective nodes. In this case, the two evidence sets ("Evidence 1" and "Evidence 2") were considered to be different episodes, and the episodic learning algorithm was run. The results are shown in Table 3.8. As can be seen by comparing the values in the table, the episodic learning adjusted the weights in the transition potential matrix to a more intuitive result based on our expectations, given the matching evidence sets.

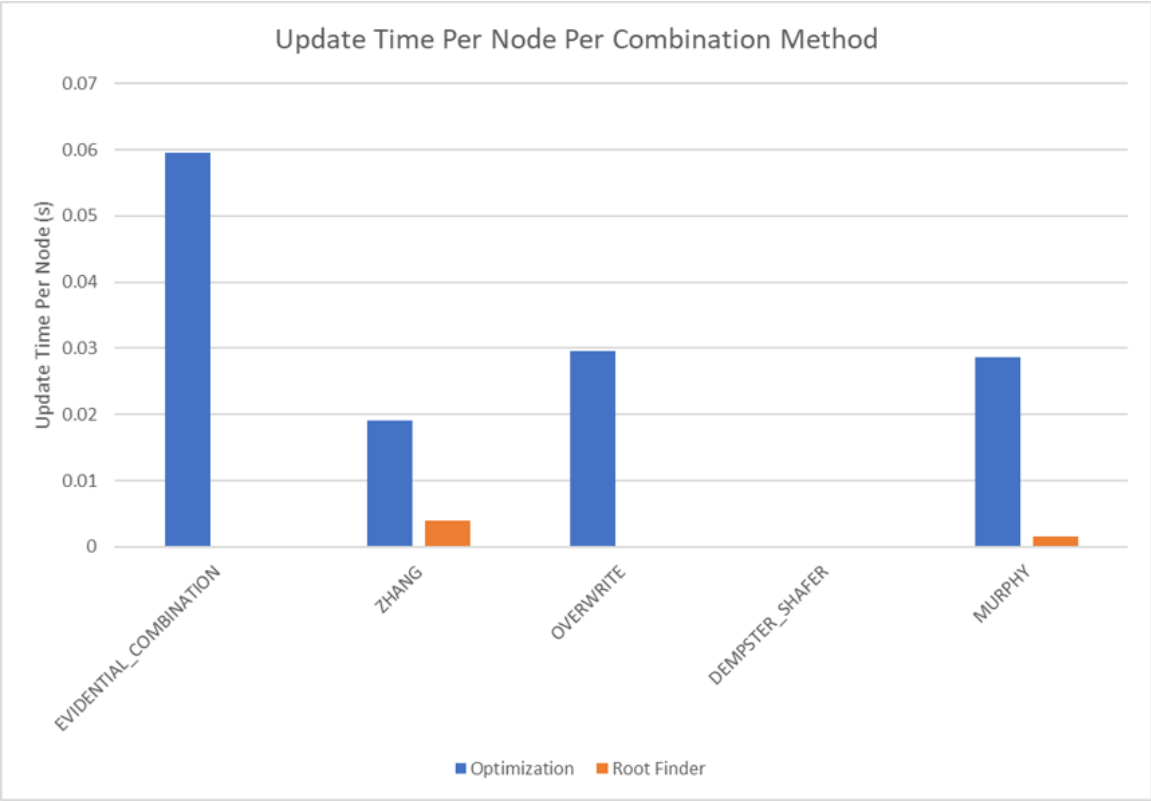

Figure 3.15: Test results for run time with a multiple parent network. The root finder primarily works for Murphy's and Zhang's combination methods, as expected. For those methods, the root finder shows at least an order of magnitude improvement in run time per node, which translates to significant improvements for larger networks.

3.7 Dempster-Shafer Network Conclusions

The new rules created for the DS network updates significantly changed how the networks function. By requiring a combination algorithm to be used at each node, similar to previous works, simultaneous evidence can be entered, allowing the transition potentials to be calculated from the evidence. This relies on the concept of consistency in the network — that transition potentials both define the influence between nodes and can be used for exact mathematical relationships. As shown in Section 3.6, propagation of evidence without learning updates to the transition potentials results in a significantly more inconsistent network. While it can be argued that the transition potentials in this case are merely measures of influence and do not require consistency, the breakdown in the mathematical structure of

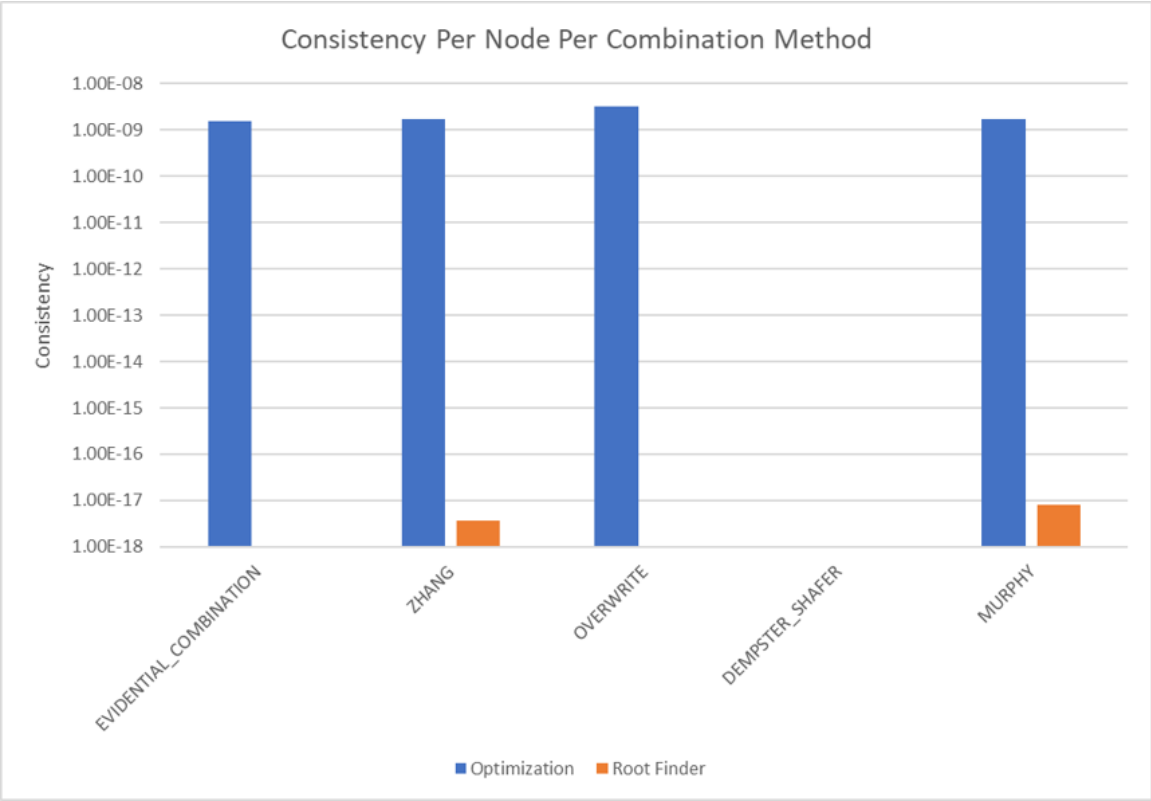

Figure 3.16: Test results for consistency with a multi-parent network. The y-axis uses a logarithmic scale. All results are within round-off error of zero, which shows perfect consistency. This result was expected given the results from the single parent case in Figure 3.11.

the network means that important properties are lost, such as the ability to reconstruct the network from partial information. There are some limiting assumptions made, primarily for multi-parent nodes. In these cases, the assumption of identical evidence inputs from the parent nodes along with the limitation to certain DS combination algorithms limits the novel algorithm developments from being used in all scenarios. However, the DS combination algorithms used can be applied to most scenarios through appropriate network definition, as discussed in Section 3.3.3. The identical evidence assumption is reasonable for many scenarios since this assumption is only made while updating the transition potentials and aligns with the initial conditions of unknown marginals and vacuous transition potential matrices. Further, that restriction only applies when using the polynomial solution; a higher order optimizer can relax that assumption. Future research trajectories include re-

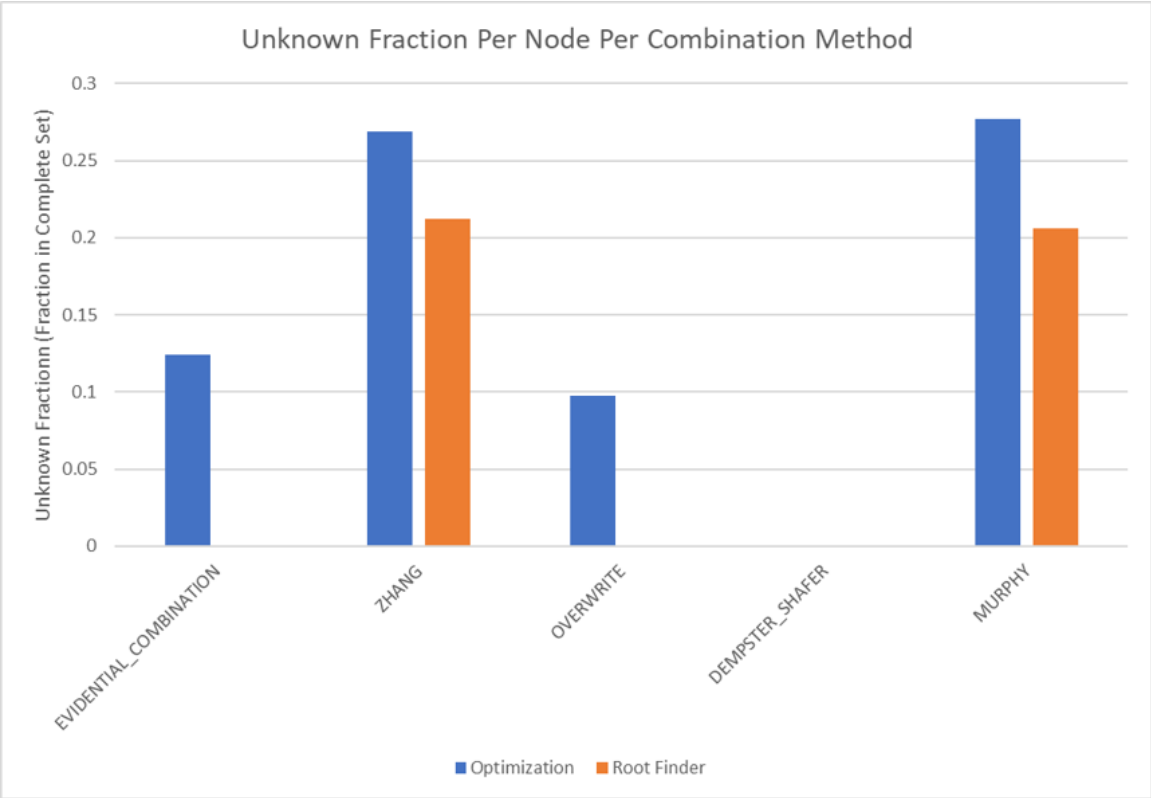

Figure 3.17: Test results for learning with a multi-parent network. The unknown fraction for the multiple parent cases are similar between the optimization and root finder methods. This result is expected, given that similar solutions should be found. Notably, significantly higher unknown fractions are found for the multiple parent cases than for the single parent cases. This difference is due to the learning method. For multiple parents, the solution method first calculates identical marginals for each parent then calculates the transition potentials per parent. Consequently, unknown information is retained significantly longer since it is duplicated multiple times.

laxing this assumption without the use of higher order optimizers.

Beyond the consistency obtained through updating the transition potentials, the learning mechanism enables starting from a completely unknown set of data to fill in the entire network. As shown in Section 3.6, learned networks with random input data results in better known networks than those with *a-priori* data provided. The *a-priori* data provided is based on assumptions of data that would be available. That *a-priori* data could serve as a starting point for learning, resulting in a better known network in all cases.

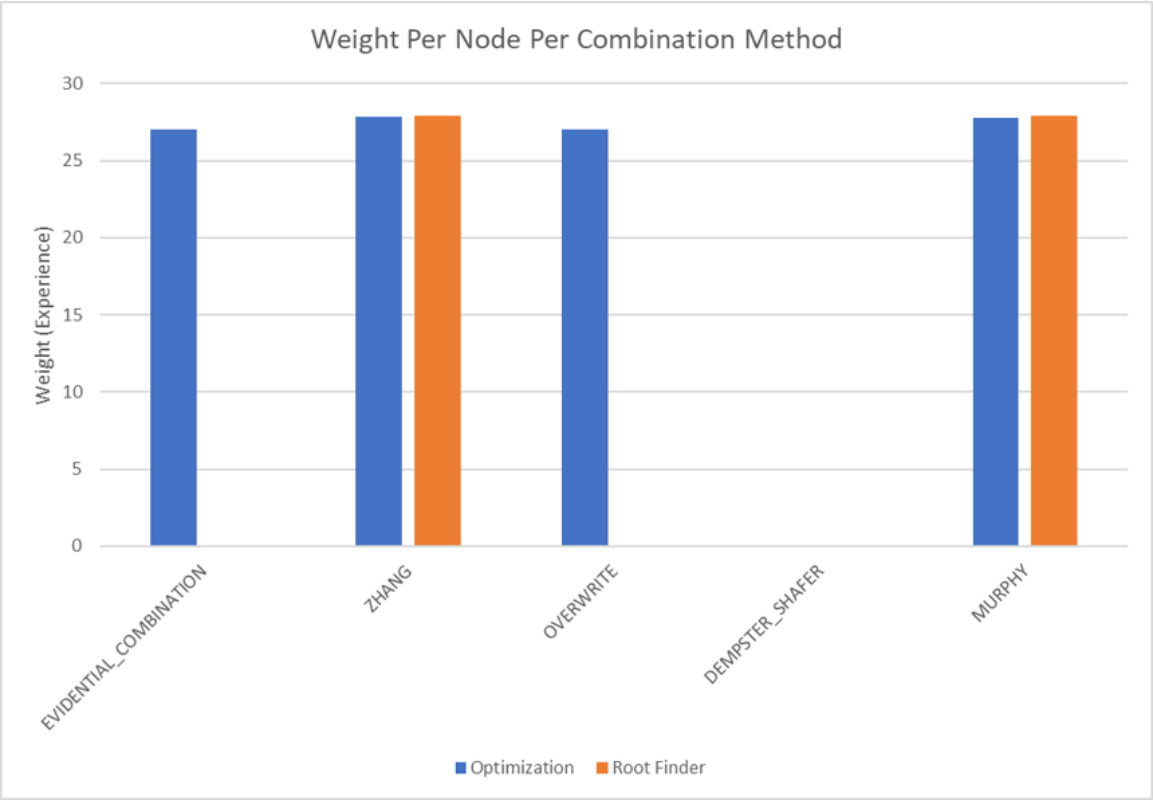

Figure 3.18: Test results for weighting with multi-parent networks. Since all cases are weighted, no deviations between cases were expected or observed. All cases show expected total weights per node of approximately 30, confirming the results from the single parent test case in Figure 3.13.

The transition update rules defined in this chapter capture relationships when the entire belief distributions can be considered simultaneously. For scenarios in which this is not the case, episodic learning was introduced to focus the transition updates on the portion of the belief distribution that was observed during the evidence, borrowing from the well-known concept of observability in control theory [102]. This novel use of episodes for updating Dempster-Shafer network transitions significantly improved how the relationships between nodes were captured, as shown in Section 4.3.2. Future research trajectories include using statistical tests to determine breaks between episodes.

Finally, the evidence weighting has a considerable impact on the updates to the network. Speed improved considerably since propagation was not required to all nodes in the net-

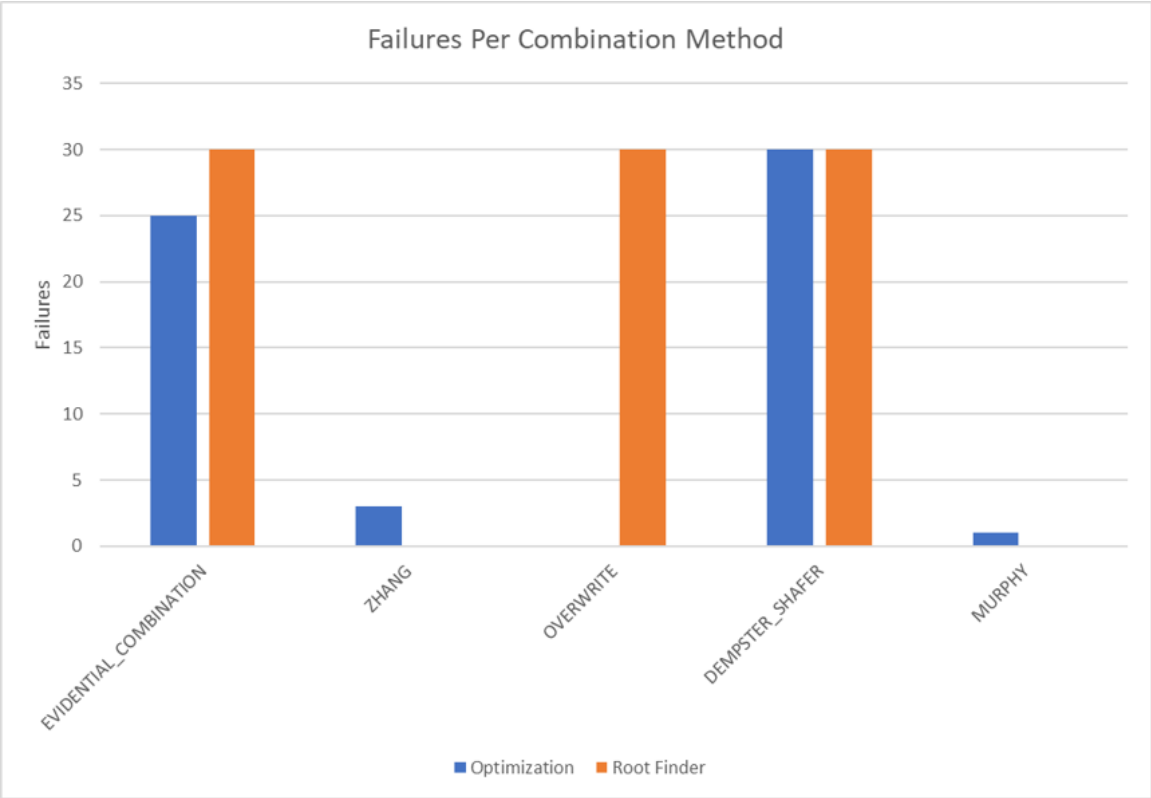

Figure 3.19: Test failures for the multiple parent cases. Three interesting effects are observed. First, the ECR method, Dempster's Rule, and the overwrite method were not expected to reliably succeed due to the random evidence sets that were not within bounds required for the reverse solver method to succeed. Indeed, these are the methods which tend to fail. Second, the root finder method is more deterministic on whether it succeeds or fails. In each set of tests, the root finder method either succeeds or fails in all tests while the optimizer can find solutions which the root finder misses. This is most evident in the overwrite method in which the root finder fails in all cases and the optimizer succeeds in all cases. However, the more practical methods, Murphy's Rule and Zhang's Rule, show better performance by the root finder.

work. Furthermore, the automatic handling of reliability of propagated evidence versus observed evidence enables more reliable results. Finally, an accurate representation of the accumulated evidence in the network enables testable results since they can be benchmarked against a consistent basis. While evidence weighting based on inference versus direct observations has already been implemented [95], Section 3.8 codifies explicit rules for calculating the weights from evidence entry in the network, creating a consistent basis for updates with clearly defined reasoning.

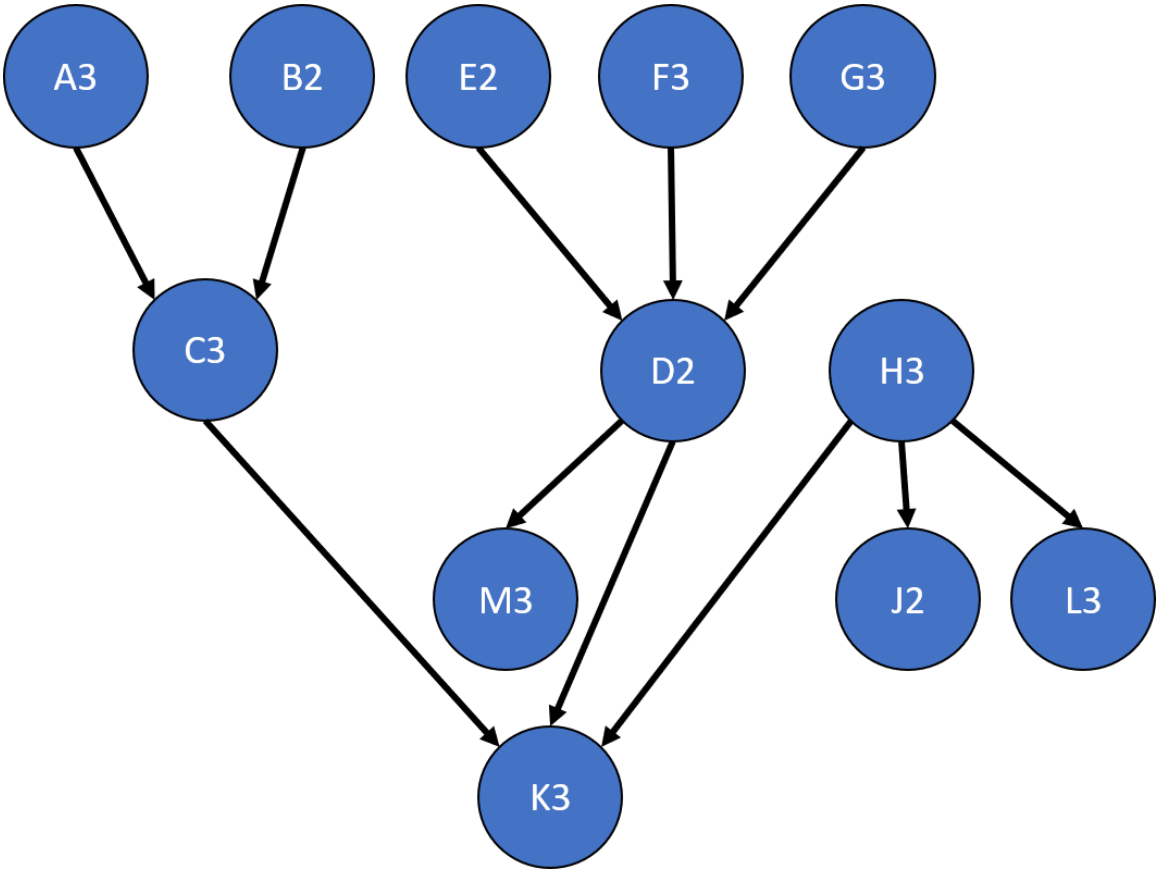

Figure 3.20: Test network used to analyze the performance of the novel Dempster-Shafer network algorithms. This example includes nodes that have both single and multiple parents. The number after the node name shows the number of θs for the node. Two and three θs nodes were used since these are the more common cases for nodes in a DS network.

One important point with these new update rules is that they do not eliminate the need for a subject matter expert (SME). Structured data is useful when there is structure of which to take advantage. That structure is typically driven by knowledge of the domain or problem — knowledge that an expert possesses. Further, designing the evidence input to enable the network to produce meaningful data still requires the knowledge of an expert. This need then raises the question of how this work applies to new domains and problems in which a subject matter expert may not exist. First, the network design process is an iterative process. In many cases the problem may start with a single DS node since relationships between θs

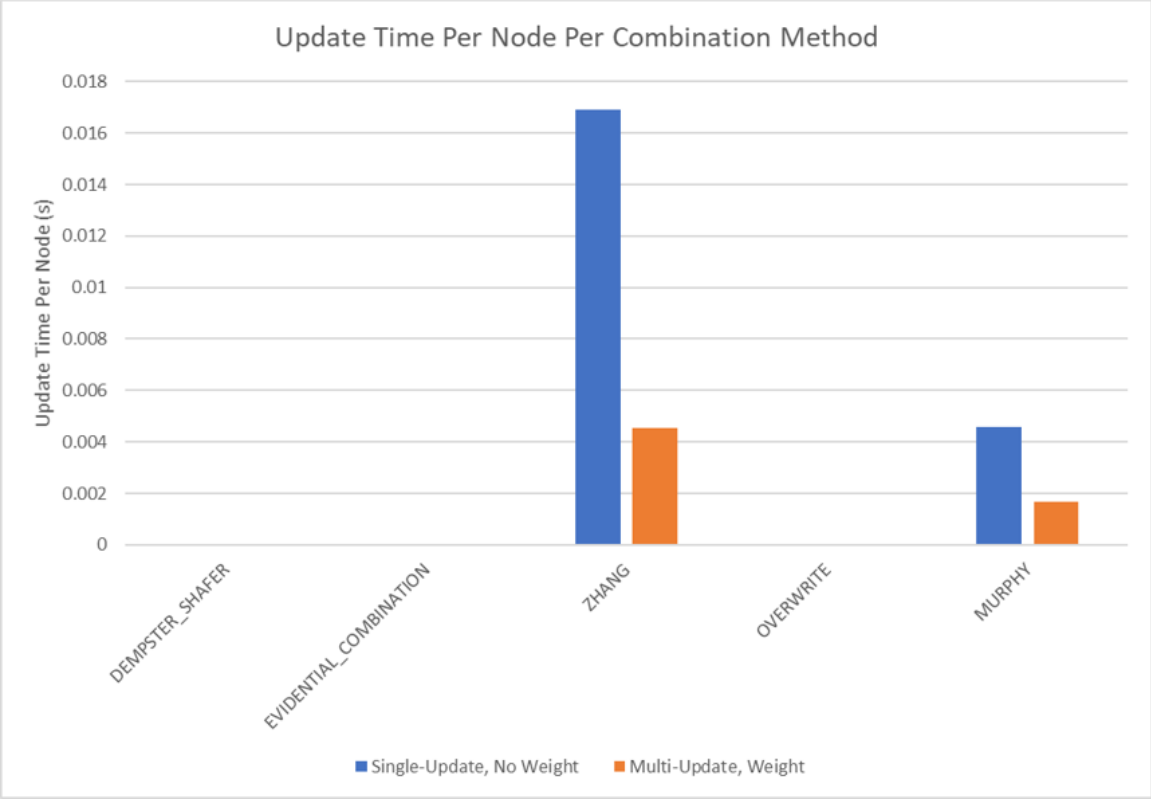

Figure 3.21: Test results for run time for a complex network. In both cases which succeeded, the weighting method significantly decreased run time, as expected. In both cases, the run time order of magnitude more closely resembles the multiple parent tests (Figure 3.15) than the single parent tests (Figure 3.10). This is expected, given that the complex network adds the additional multi-parent calculations. These results also suggest that the root finding method for multi-parents still dominates the single parent solution method.

are not known *a-priori*. The network is constructed iteratively as evidence combination shows which θs do not interact or have directed, conditional relationships. Second, the enumerated θs can be added over time, either through the use of an open-world definition initially to suggest when the enumerated θs are not sufficient, or by observing the resulting marginal distributions and determining whether the DS node is insufficient to choose between multiple θs, suggesting there are either overlaps or no θs are sufficiently granular to be the "correct" choice. All of this analysis is still predicated on reasonable choices of evidence entry. This part tends to rely on a DS SME — i.e. designing a problem appropriately for DS. This part requires less domain expertise and more understanding of the appropriate

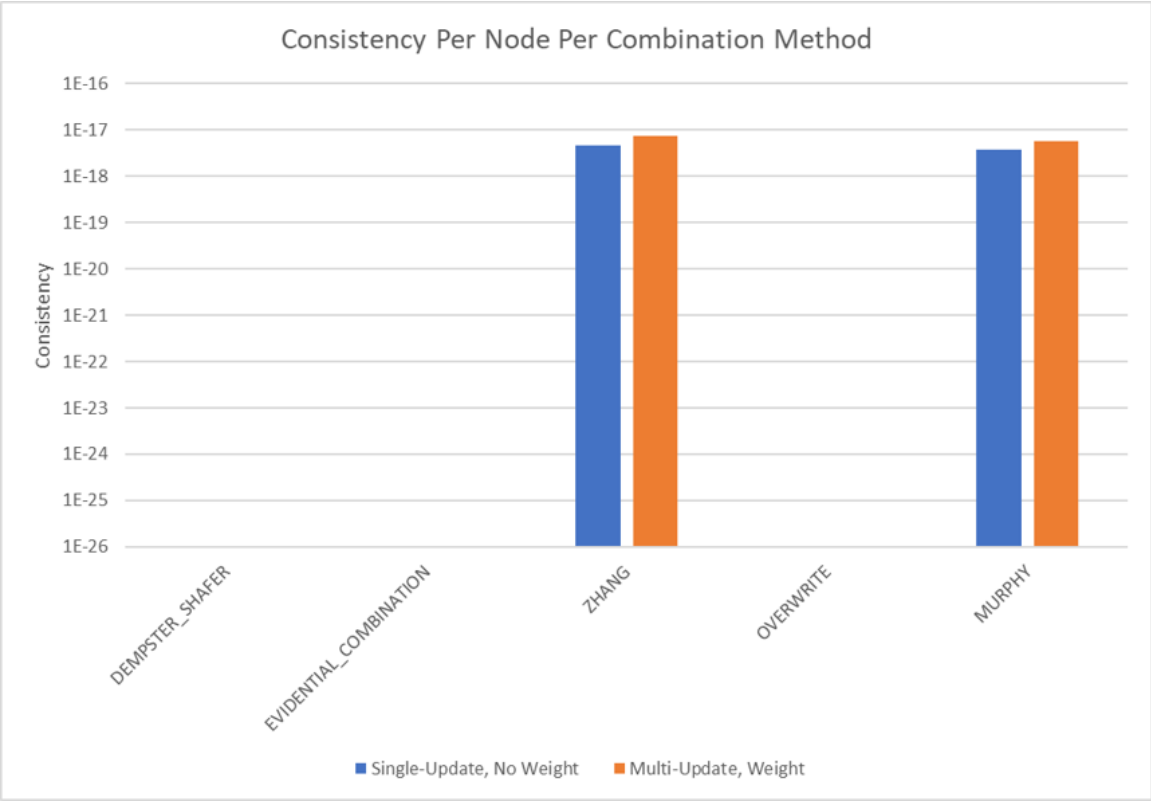

Figure 3.22: Test results for consistency for a complex network. For all tested cases, the consistency is effectively zero with round-off error, demonstrating perfect consistency in line with the results from the single and multi-parent solutions (Figures 3.11 and 3.16, respectively).

use of mapping to unknown and ambiguous evidence as well as the appropriate weighting of evidence and windowing of decision data, thereby controlling how the DS network converges to a solution. Typically, this work can also be iterative, often starting with higher unknowns and ambiguities and becoming more precise as an understanding of the scenario decision dynamics is learned over time.

Together, these rules realize a capability which was envisioned in the 1980s [17], extended through multiple works in the 1990s and 2000s, and brought to fruition through this work. While valuation networks provide an enormously capable system for understanding and evaluating knowledge of the world, this work enables starting from unknown or limited information and learning relationships without requiring continuous significant

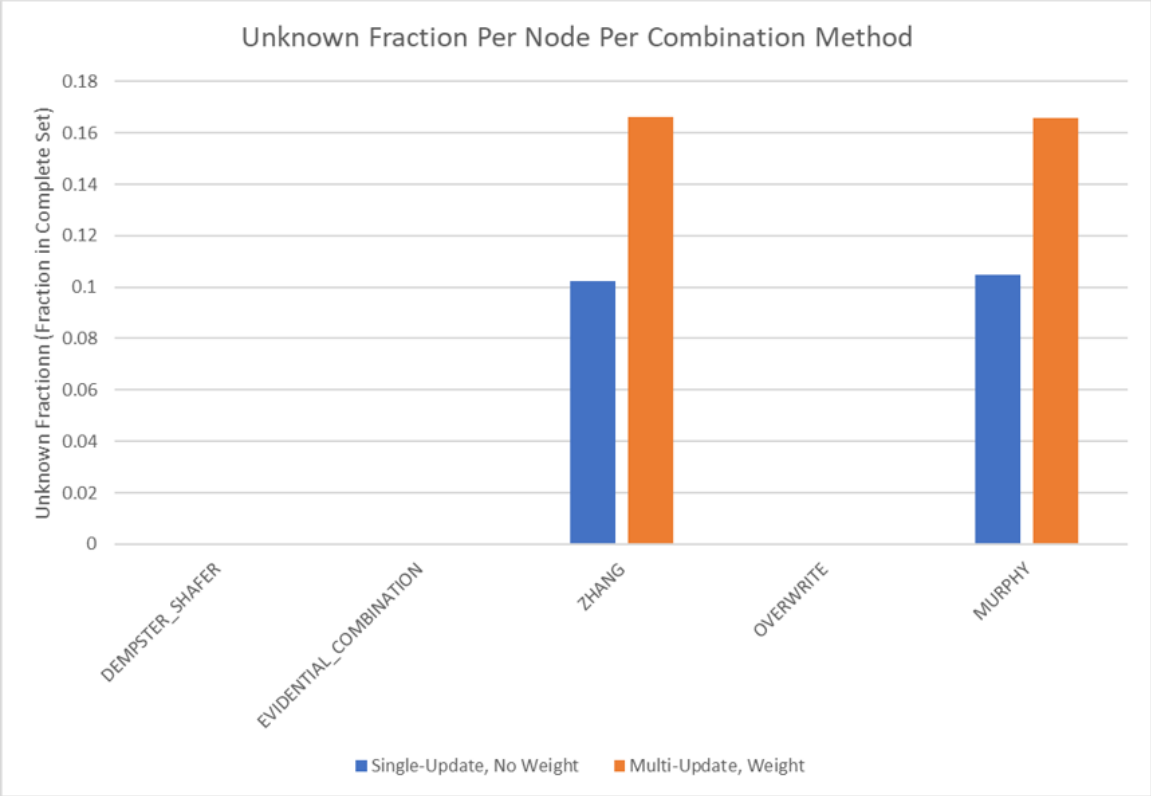

Figure 3.23: Test results for learning for a complex network. There are two points of interest here: (1) The weighted, multiple update method does display a higher unknown fraction. This opposes the results seen in the single-parent tests (Figure 3.12), suggesting that the multiple parent solution method fares less well when dealing with weighted data; (2) The unknown fraction is between the single parent tests (Figure 3.12), and the multi parent tests (Figure 3.17), which is expected, given that the complex network is a combination of the previous networks.

subject matter expert's inputs — a capability which allows reasoning about the world to progress quickly and with reduced guidance from experts.

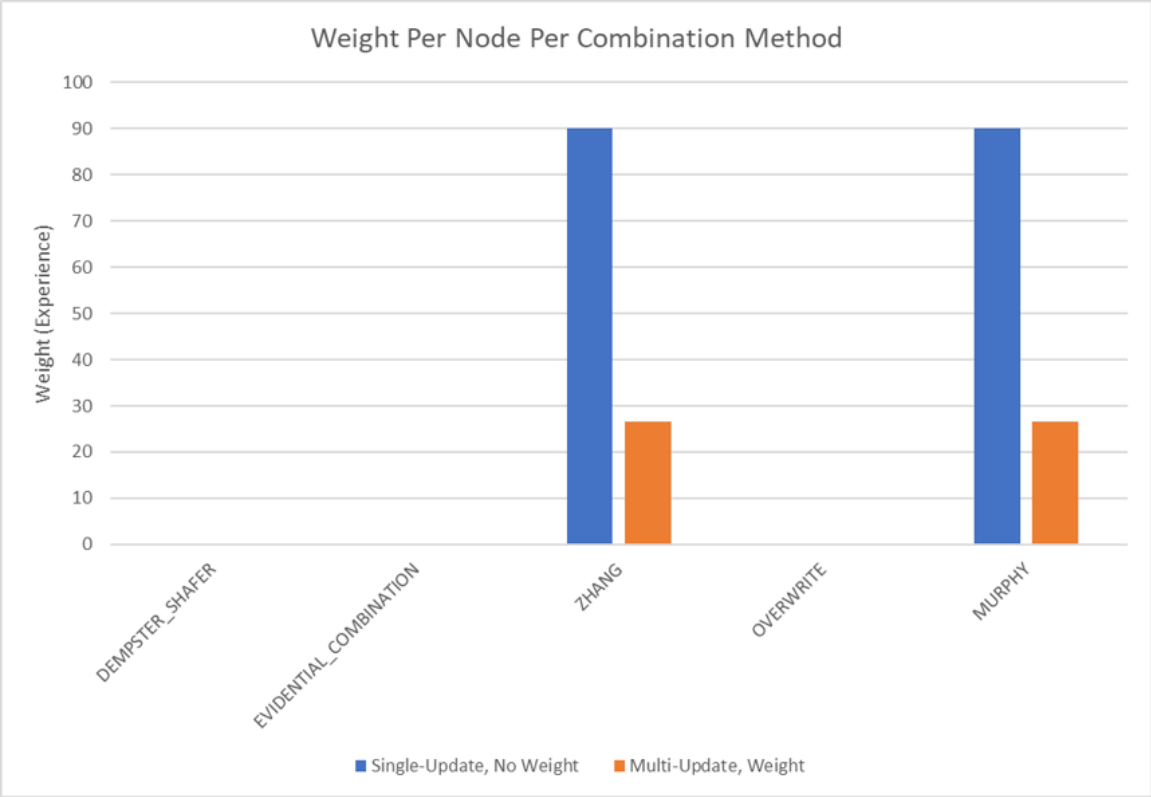

Figure 3.24: Test results for weight for a complex network. This result is consistent with the results seen previous in Figures 3.13 and 3.18, suggesting that the weighting results obtained in this section can be extended to DS networks of arbitrary complexity.

Table 3.7: Episodic learning test. The baseline without episodic learning started with unknown information (all marginal masses in the complete sets). Both evidence sets were added and combined via Murphy's Rule [19] into their respective nodes. The transition potential matrix update algorithm was run against the resulting node marginals after each evidence update. The second test applied the "Evidence 1" sets to the appropriate nodes and ran the transition potential matrix update algorithm against the resulting node marginals. The node marginals were then reset to the unknown state, and the "Evidence 2" sets were applied to the appropriate nodes. The transition potential matrix update algorithm was run against the resulting node marginals again to incorporate the second episode into the resulting transition potentials matrix.

Data Set	Option_A	Option_B	(Option_A, Option_B)	Option_C	Option_D	(Option_C, Option_D)
Evidence 1	0.9	0.08	0.02	0.9	0.1	0.0
Evidence 2	0.05	0.9	0.05	0.1	0.7	0.2

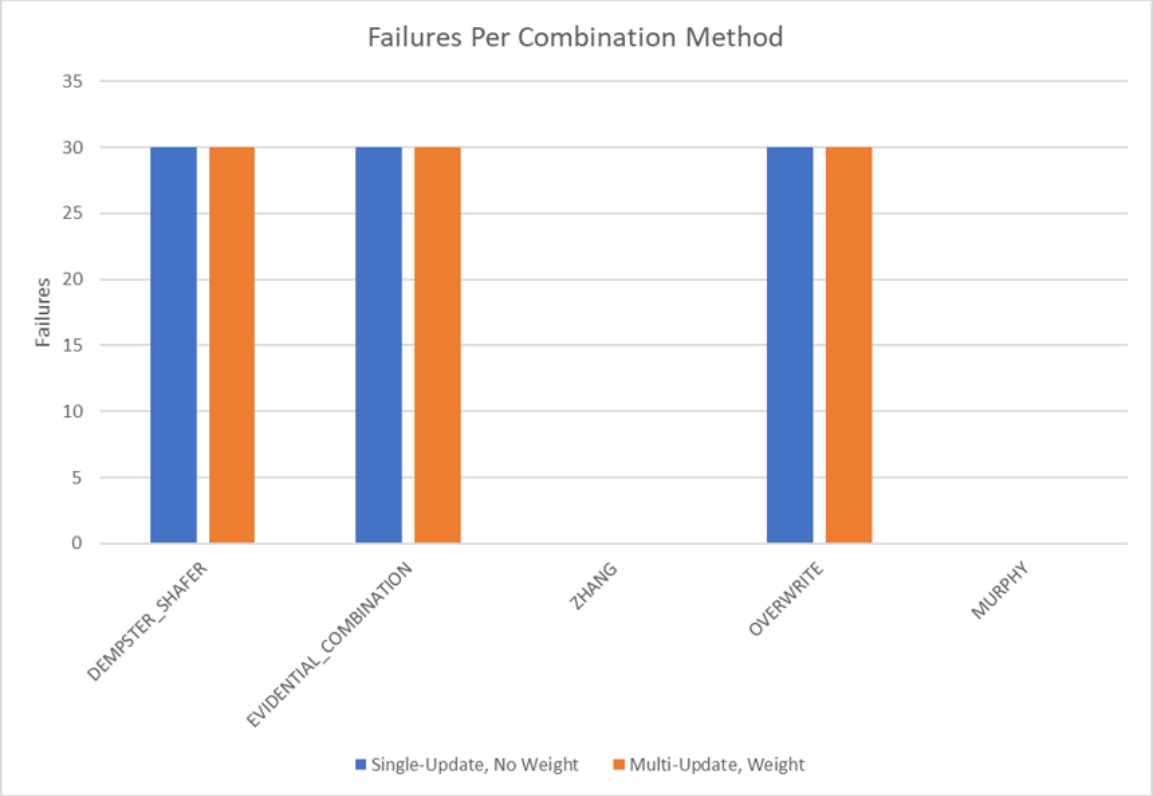

Figure 3.25: Test results for failures for a complex network. The root finder is used for both tests, with these results showing that the weighting method has no effect on whether the root-finding method is able to find a feasible solution.

Table 3.8: Episodic learning results. The baseline without episodic learning started with unknown information (all marginal masses in the complete sets). Both evidence sets were added and combined via Murphy's Rule [19] into their respective nodes. The transition potential matrix update algorithm was run against the resulting node marginals after each evidence update. The second test applied the "Evidence 1" sets to the appropriate nodes and ran the transition potential matrix update algorithm against the resulting node marginals. The node marginals were then reset to the unknown state, and the "Evidence 2" sets were applied to the appropriate nodes. The transition potential matrix update algorithm was run against the resulting node marginals again to incorporate the second episode into the resulting transition potentials matrix.

Without Episodic	Option_A	Option_B	(Option_A, Option_B)
Option_C	0.552	0.455	0.821
Option_D	0.448	0.329	0.179
(Option_C, Option_D)	0.0	0.216	0.0
With Episodic	Option_A	Option_B	(Option_A, Option_B)
Option_C	0.727	0.0	0.233
Option_D	0.129	0.703	0.336
(Option_C, Option_D)	0.143	0.297	0.431

CHAPTER 4

AUTONOMOUS CAR DECISION - TRAFFIC LIGHT SCENARIO

4.1 Scenario and Metrics Definition

Autonomous car development has become prolific as more manufacturers, such as Tesla [103] and Waymo [104], move to bring self-driving vehicles to market. A common decision faced by driver — human or computer — is whether to slow down early or late when approaching a red light. Prior research shows that the maximum comfortable deceleration rate for automobiles is approximately $3.4\frac{m}{s^2}$ [21]. Drivers assuming the light will change to green soon may wait until closer to the light to slow down, with deceleration rates often approaching that maximum deceleration. A separate research project used GPS modules in vehicles driven around Atlanta, Georgia to measure the deceleration rates [21]. This research found that the deceleration rates were non-constant, typically with lower deceleration rates at the beginning and end of the maneuver and highest in the middle of the maneuver. Additionally, the deceleration took place over a longer distance than the maximum rate deceleration maneuver.

Using the prior research as a baseline for slowdown profiles, the Dempster-Shafer (DS) network was tested given the scenario shown in Figure 4.1. The following simplifying assumptions were made in this scenario:

- The traffic light is open loop controlled: each light is timed without using mechanisms to detect the presence of cars. While this assumption constrains the traffic lights to which this analysis applies, there are traffic lights that follow this assumption, so it is not unreasonable.

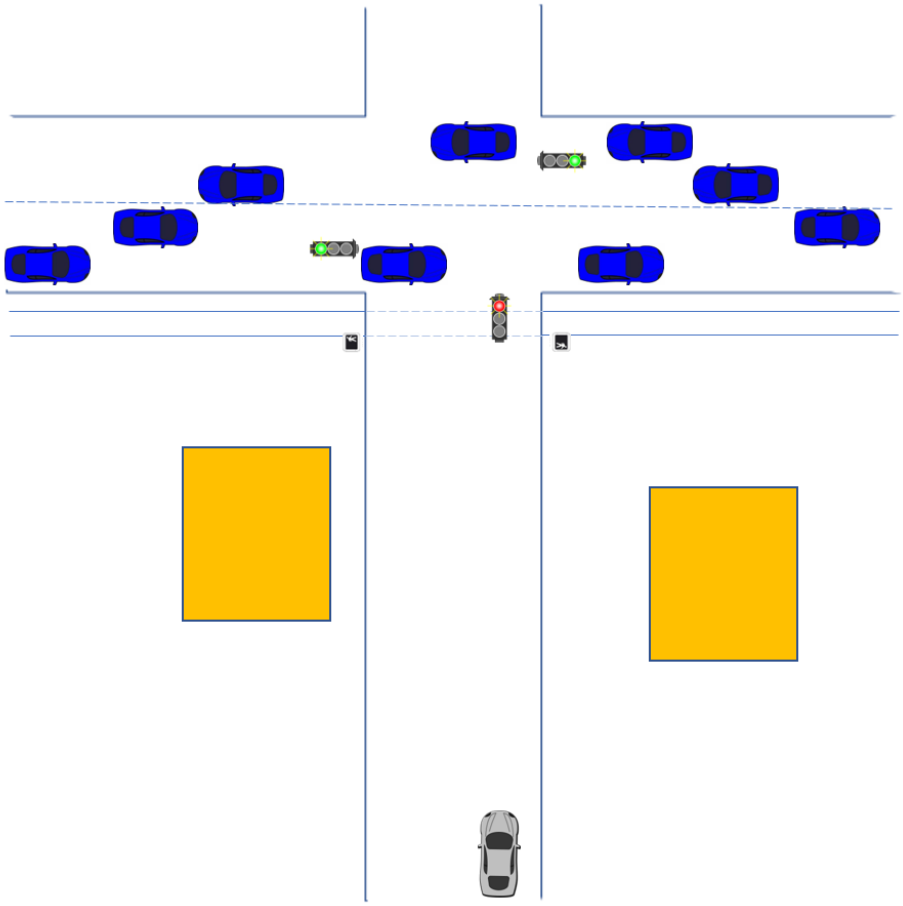

Figure 4.1: The layout for traffic signal scenario which is representative of timed four-way lights without left turn signals. The grey car is approaching a red light intersection and estimating how long until the light turns green to determine whether to slow the car. Visibility is limited due to buildings and other obstructions. The cross-walk signal may be visible before the intersection. The cross-traffic light is not visible to the grey vehicle and must be estimated. Cross traffic density and speed is variable in the simulation and is estimated by the grey vehicle.

- The light is a 4-way light without lights for left turns.
- Vehicles from every direction approach the light at the same speed limit. This assumption is without loss of generality since evidence compares observed speeds to expected speeds, and the expected speeds can be modified to match different speed limits.
- Individual vehicles do not have to be modeled. Evidence inputs to the DS network

are aggregated, thus allowing the modeling to be aggregated as well.

- Per one driver, an expressed goal is to stay at approach speed as long as possible in hopes that the light changes while maintaining the ability to slow down and stop before the intersection if the light stays red. This goal provided a reasonable decision criteria to use when evaluating whether the DS network performs better than baseline deceleration profiles. Since this goal would typically result in choosing the maximum rate deceleration profile, and additional goal was added to minimize wear on the vehicle.

- Drivers pick pick one of the deceleration profiles listed above depending on their estimation of how long the light will be red. That choice can change as the driver approaches the intersection.

- Drivers can change their speed without maneuvering around other vehicles. While this is an over-simplification for most traffic scenarios, the complexity of dealing with multiple drivers making inter-related decisions obscures the analysis this scenario was designed to test.

Given the above assumptions, a simulation was developed to test the DS decisions against a baseline deceleration profile. Four deceleration profiles were chosen. A fifth profile, minimum constant deceleration, was evaluated, then removed since this profile showed no advantages over the other profiles; it took longer to slow down, did not produce significantly less deceleration on the vehicle, and is not shown by research to be representative of realistic drivers. The drivers were compared by starting each scenario with two drivers: a DS-informed driver, and a driver with one of the baseline profiles. By providing each driver with an identical scenario each time, metrics could be evaluated based on the difference between each of the drivers. Note that each simulation run presented a different scenario due to changes in approach speed, cross-traffic density and speed, visibility, and access to other evidential inputs. Detailed initial conditions are available in Appendix D.

- Variable rate deceleration profile: this profile uses the GPS-tagged research described above.
- Maximum deceleration profile: this profile assumes the driver waits as long as possible before changing speeds, then slows down quickly.
- Coin Toss: this profile switches between the above two profiles with a 50-50 probability and was included to determine whether a DS-informed driver could perform consistently better than simply flipping a coin.
- Bayesian: this profile switches between the above first two profiles based on a Bayesian evaluation of the traffic light state using the same observations available to the DS-informed driver. The Bayesian Belief Network (BBN) code used for this test was obtained from GitHub [22].

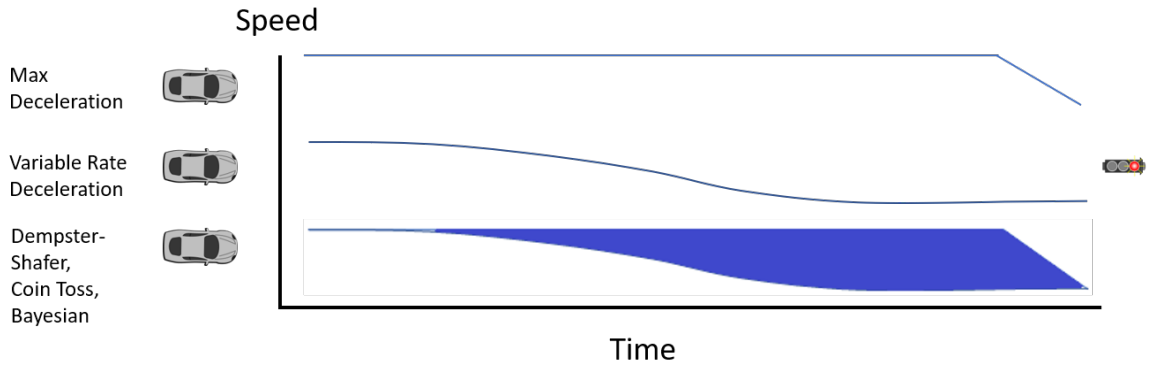

Figure 4.2: The deceleration profiles for the traffic signal scenario. As the grey car approaches the red light, the two naïve deceleration profiles are max deceleration and variable rate profiles. The alternate profile is anything between those two naïve profiles based on Dempster-Shafer analysis, a coin toss, or a Bayesian evaluation to determine which profile to follow at each decision point.

These profiles are shown in Figure 4.2. In order to compare the DS-informed driver with the four profiles given, the following metrics were chosen and evaluated based on the difference between the DS-informed driver and each of the baseline profiles. Figure 4.3 compares the baselines to show the available gap that the DS-informed driver can use to improve on these metrics. This figure shows the maximum and minimum values of the error bars,

demonstrating the full extent of the range for this scenario. An interesting point is that choosing a specific baseline over the course of the entire scenario without analyzing the traffic light does not result in the driver consistently maintaining a higher speed when the light changes. However, due to the range of differences between the baselines with regards to the speed metric, there is an ability to maintain a higher speed overall if intelligent choices are made.

- Speed when light changes: if at least one vehicle (the DS-informed driver or the baseline driver) has not come to a complete stop when the light changes, the difference in speed is calculated.
- Distance when light changes: if at least one vehicle (the DS-informed driver or the baseline driver) has not come to a complete stop when the light changes, the difference in distance from the intersection is calculated.
- Time until light changes: if both vehicles (the DS-informed driver and the baseline driver) have stopped before the light changes to green, this metric calculates the difference in time that the drivers have to wait at the intersection (how closely they timed their stop to when the light would change).
- Wear: This metric is defined as $\int_{t_0}^{t_f} \frac{decel}{max \quad decel} * dt$ where t_0 is the time at which the vehicle starts decelerating and t_f is the time at which the vehicle stops decelerating. It has been shown that there is a relationship between harder braking (i.e higher deceleration rates) and higher wear on the tires and brakes. While the relationship is not necessarily quadratic, the quadratic relationship shows a clear difference between consistently higher deceleration rates and the variable rate profile, enabling easy evaluation of the effect of braking on the vehicle.
- Number of times the vehicle entered the intersection: this was a potential metric that was removed in favor of choosing "aware" drivers: drivers that recognize when they will enter the intersection and brake at the maximum deceleration rate to prevent it.

By making this choice to use “aware”, this metric is rolled into the “Wear” metric with the additional maximum deceleration.

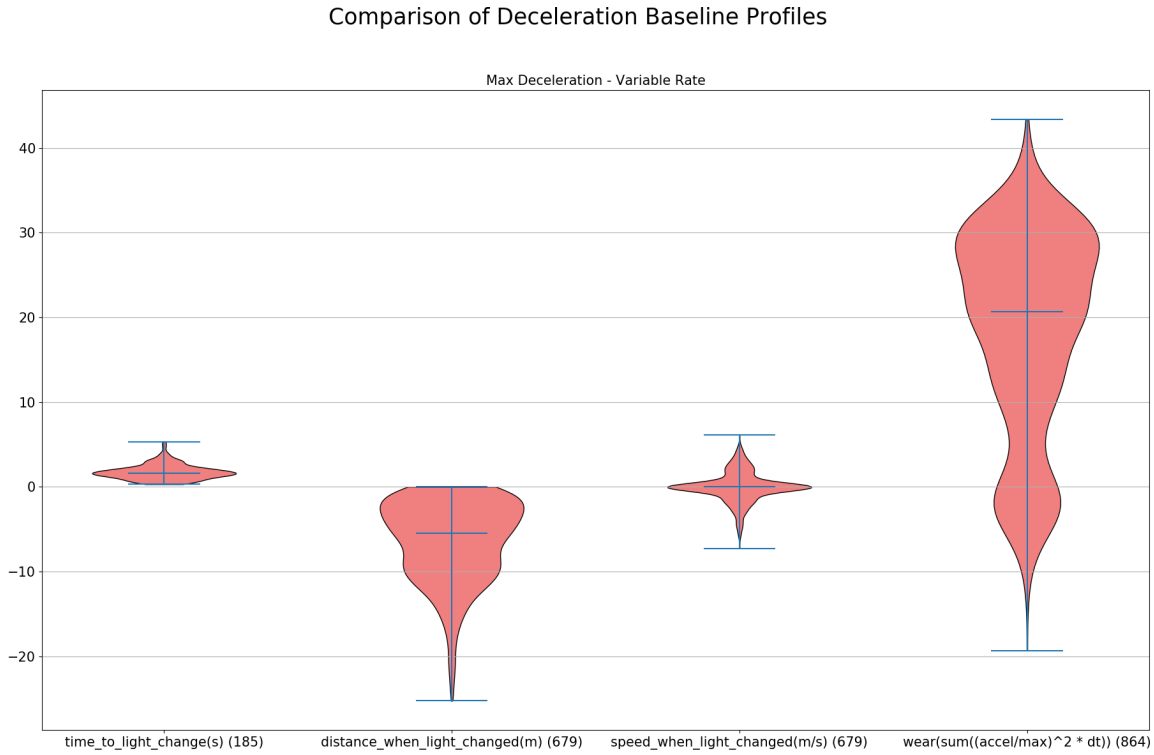

Figure 4.3: The comparison between the Maximum Deceleration and Variable Rate Deceleration Profiles using the stated metrics. These values are calculated by subtracting the Variable Rate Profile results from the Maximum Deceleration Profile results. Thus, a value greater than zero means that the Maximum Deceleration Profile resulted in a higher value in that metric than the Variable Rate Deceleration Profile. Y-axis values for each metric are in the units specified by that metric’s label. Clearly, the largest gap is in wear, while the speeds tend to even out over time for the given scenario. Since the Dempster-Shafer-informed driver switches between these baselines, this graph shows the potential improvement over either baseline by the Dempster-Shafer-informed driver.

4.2 Dempster-Shafer Network and Decision Design

Given the scenario defined in Section 4.1, the DS network shown in Figure 4.4 was designed to evaluate the scenario. This network does not take into account varying environmental conditions, so those are assumed to be constant throughout the scenario. Likewise, no interference was assumed between drivers. With these assumptions, the network is sufficiently

complex to test all the capabilities of the network development in this paper, including nodes with single and multiple parents, while sufficiently simple to explain concisely. Expert information was used to determine the relevant relationships in the network. Recall that no link between two nodes does not mean that there is not a relationship between those nodes. Rather, it means that there is negligible direct effect between those nodes. Thus, cross traffic density has negligible effect on the time until the traffic light turns green. In reality, the time until the traffic light turns green is not the only reason whether the evaluating vehicle can cross the intersection. The intersection must be clear, and higher density cross traffic is more likely to result in a congested intersection even after the light has changed. However, those relationships are considered negligible for this scenario in comparison to the relationships defined.

All nodes are assumed to be observable during network training. This assumption is reasonable given that a trainer could sit at an angle relative to an intersection that allows that observer to see or compute all information present in the network nodes. Comparisons were made between the effects of observing the cross-traffic light and not observing the cross-traffic light during training; those comparisons are detailed in Section 4.3.2.

Evidence input design, in addition to network design, is important in DS analysis. Thus, the functions which convert observations to evidence must be defined, and these impact the resulting output and decision criteria. For this analysis, the evidence input for the "Time to Green" node will be discussed. Figure 4.5 shows the function used to convert observations of the time until the light changes to green into categorical evidence inputs. For this function to be evaluated, maximum short time and maximum medium time must be discussed, along with the fraction for the complete set — the unknown information. Maximum short time was defined as $\frac{1}{3} * \frac{Variable\_rate\_slowdown\_distance}{current\_speed}$. Maximum medium time was defined as $\frac{1}{3} * \frac{Variable\_rate\_slowdown\_distance}{current\_speed * \frac{3}{4}}$. These definitions were chosen to abstract away distance, connect to concepts that could be observed in a real-world scenario, and provide reasonable

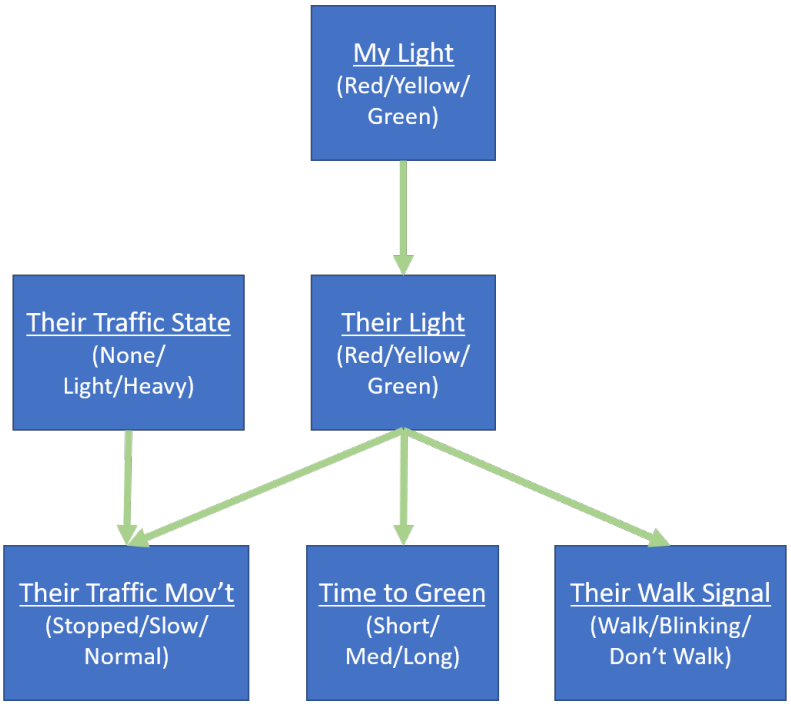

Figure 4.4: The Dempster-Shafer network used to evaluate the traffic light scenario. Green links represent the relationships between nodes in the direction of effect. For example, the state of "Their Light" affects the state of "Their Traffic Movement". The reverse, in general, is not true, although inferences can be made if traffic movement is observed. The "My Light" node is included primarily to ensure that onces the light changes, the network will immediately update the time until the light changes to green.

values for decision criteria.

Using the evidence inputs described above, decision criteria were defined to balance uncertainty with the driver's goals stated previously (maintain speed as long as it is possible the light with change to green before the driver enters the intersection, retain the ability to stop safely if the light does not change, and minimize wear on the vehicle). In order to translate the scenario goal into decision criteria, Figure 4.6 was developed.

Per the decision criteria chosen in Section 4.1 and the decision criteria concepts defined in Figure 4.6, the appropriate decision criteria was chosen to be a combination of the "High Possibility, Belief" limit in Figure 4.6 for the "Short" θ in the "Time to Green" node and the "Low Belief" limit in Figure 4.6 for the "Medium" and "Long" θs in the "Time to

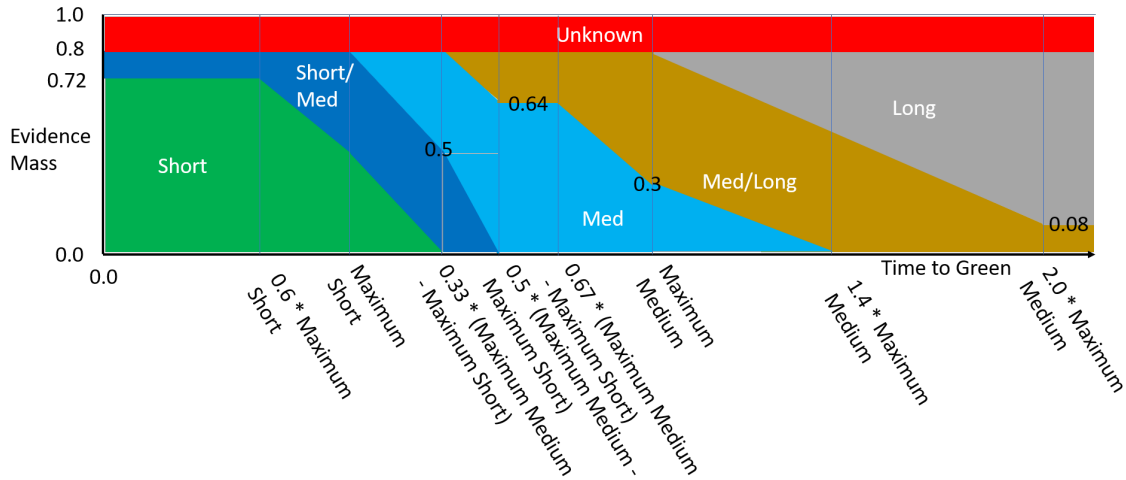

Figure 4.5: The figure represents the function for computing evidence input for the "Time to Green" node from network training observations. This figure is read as follows: given a observed time until the light changed to green on the x-axis, the evidence input for the "Time to Green" node can be calculated from the belief mass assignments along the y-axis. For example, if the scenario takes zero seconds for the traffic light to change to green, then the evidence input is 0.72 **Short,** 0.08 **(Short, Medium), and** 0.2 **unknown, which is equivalent to (Short, Medium, Long). This function should introduce uncertainty since stark conflict in evidence inputs results in rapidly changing decision outcomes. Thus, as the function approaches values between clear situations of long, medium, and short, the majority of weight is placed into ambiguous evidence inputs, enabling the Dempster-Shafer combination method chosen to combine the evidence and return a reasonable outcome.**

Green" node. The table defining the decision criteria is shown in Table 4.1. The original values highlighted a wide gap between the acceptable belief for the "Short" θ in the "Time to Green" node and the acceptable possibility of that θ. In practice, however, it was found that for reasonable windows of evidence evaluation, the gap closed considerably, leading to the updated criteria. A more detailed evaluation of this difference is given in Section 4.3.1.

4.3 Evaluation and Test

Results of the scenario evaluation are shown in Figures 4.7 – 4.10. These are direct comparisons between drivers in each simulation. In all cases, the values reflect the difference of the

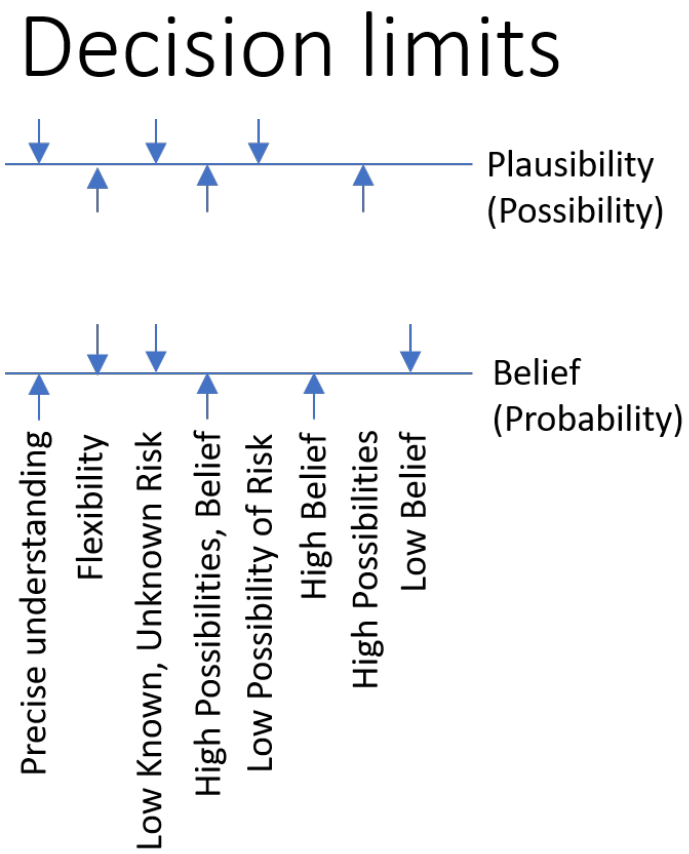

Figure 4.6: Decision criteria defined for Dempster-Shafer analysis. Precise Understanding means that the decision-maker requires little-to-no unknown. Flexibility means that the decision-maker requires significant ambiguity or unknown in the system. Low Known, Unknown Risk means that the decision-maker requires both the probability of a particular risk as well as the maximum possibility of that risk to be low. High Possibilities, Belief means that the decision-maker requires that there is a strong belief that the value under consideration is true, and the possibility of that value being true is very high; this case could be applied to the stock market. Low Possibility of Risk is used when the decision-maker is only concerned with the maximum possibility of a risk; this limit could be applied when the impact of the risk being realized is too high to accept, so the possibility must be minimized. High Belief is used when only the probability of the situation is important to the decision-maker; this choice is a typical Bayesian approach. High Possibilities is used when only a the possibility of a situation is of interest to the decision-maker; this criteria could likely be used in gambling situations. Finally, Low Belief is used when only the probability of the situation is important to the decision-maker; this limit is, again, a typical Bayesian approach.

Table 4.1: The decision criteria translated into numerical limits that can be evaluated by the Dempster-Shafer network. Any "greater than" limit means that the associated Belief or Plausibility value must be greater than that limit to meet the criteria. For example, the belief in the "Short" θ must be greater than 0.2 to meet the decision criteria. Likewise, any "less than" limit means that the associated Belief or Plausibility value must be less than that limit to meet the criteria. This table presents the original values chosen, which were then updated to align with the values that the network produced, as described in Section 4.3.2.

Criteria Set	Limit Application	Short	Medium	Long
Original	Belief	≥ 0.2	≤ 0.50	≤ 0.2
	Plausibility	≥ 0.9		
Updated	Belief	≥ 0.2	≤ 0.75	≤ 0.4
	Plausibility	≥ 0.25		

DS-informed driver minus the baseline profile driver. Expectations were as follows:

- The DS-informed driver would end up at a slower speed, farther from the intersection, and with significantly less wear than the max deceleration driver, but the time to change would be less for the DS-informed driver when both vehicles have to stop.
- The DS-informed driver would end up at a higher speed, closer to the intersection, and with more wear than the variable rate profile driver, and the time to change would be higher for the DS-informed driver when both vehicles have to stop.
- The DS-informed driver would end up at a higher speed, closer to the intersection, and with less wear than the coin toss profile driver, and the time to change would be less for the DS-informed driver when both vehicles have to stop.

The first comparison is between the DS-informed driver and the max deceleration profile, shown in Figure 4.7. There are several points to be observed in this comparison. First, the wear is almost always lower for the DS-informed driver than for the max deceleration driver, with a large maximum difference. Note that the wear in "ds_maintained" case has a higher maximum difference than the wear in the "ds_not_maintained" case, suggesting that remaining at speed at appropriate times can reduce wear more significantly. In all cases, the distance when the light changed metric is greater for the DS-informed driver than for

the max deceleration profile, which is expected, given that if at least one vehicle was still moving when the light changed, the max deceleration driver should have always been closer or the same distance to the light as the DS-informed driver. The speed when light changed was lower for the DS-informed driver for "ds_not_maintained" cases, which is expected, but was reasonably consistent with the max deceleration profile values for the "ds_maintained" cases. This result suggests that the DS-informed driver could match the speed of the max deceleration profile driver while still achieving better wear patterns. The time to change metric was always less for the DS-informed driver, which is expected, given that the DS-informed driver never slowed later than the max deceleration driver. It was expected there there would be a significant number of simulation results with zero difference because there were many simulation runs in which the maximum deceleration profile was the best choice, and the DS-informed driver should choose the maximum deceleration profile consistently in those cases. Beyond the zero difference cases, the wear distribution shows a consistent improvement over the max deceleration profile, and the speed shows some improvements as well; slowing down early then maintaining speed led to many cases in which the DS-informed driver had a higher speed than the maximum deceleration profile driver when the light changed to green.

The second comparison is between the DS-informed driver and the variable rate deceleration profile driver. Again, the first point to observe is the wear pattern — the DS-informed driver almost always has higher wear than the variable rate deceleration driver. This result is expected given that the variable rate deceleration profile provided the least wear. The question is whether the higher wear is worth the choice. Looking at the speed comparison, the speeds are effectively the same between drivers when the light changed. In the "ds_not_maintained" case, the DS-informed driver appears to have higher speeds, but the difference is mostly negligible. The primary difference when at least one vehicle is still moving is the distance to the light. The DS-informed driver consistently is closer to the light when it changes, with values ranging between 0m and approximately 12m. Finally,

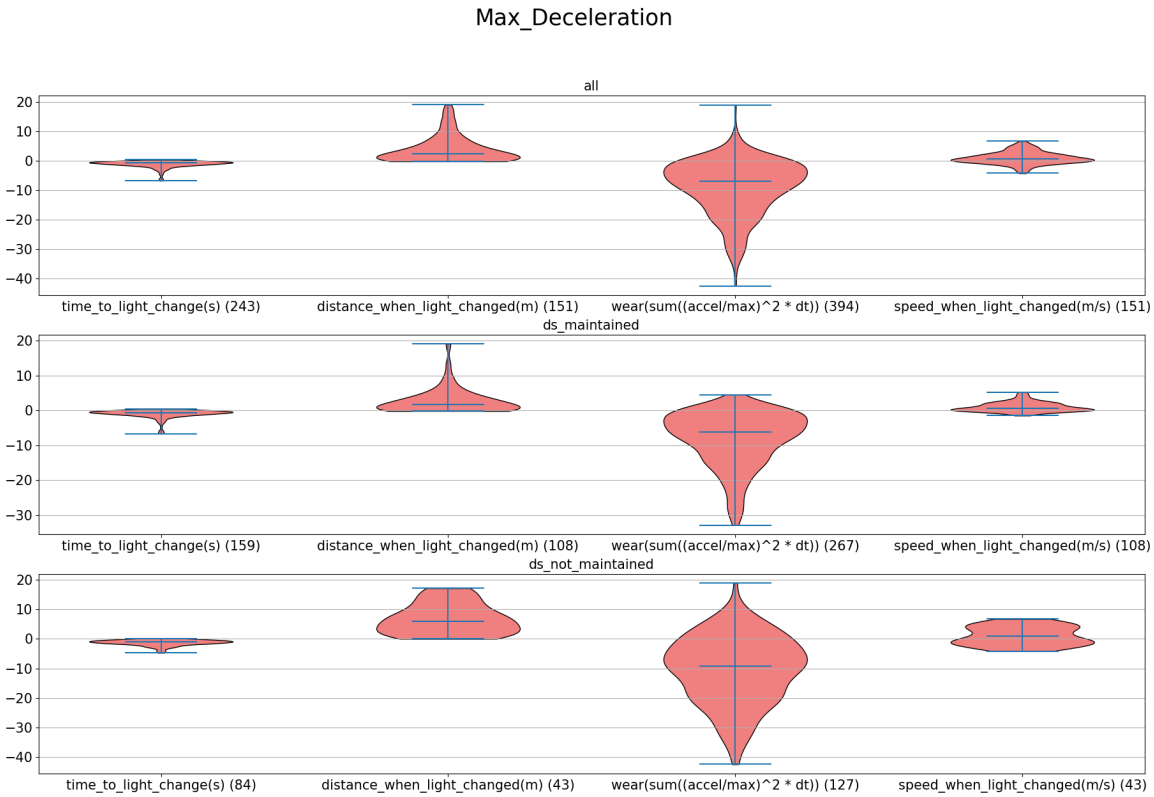

Figure 4.7: Comparison between the Dempster-Shafer-informed driver and a driver always following the maximum deceleration profile. Distributions are obtained by subtracting the results for the baseline profile driver from the DS-informed driver results for each simulation run. The three plots are All (include all data), "ds_maintained" (includes data in which the Dempster-Shafer evaluation told the driver to stay at speed at least 25% of the decisions), and "ds_not_maintained" (includes data not included in the "ds_maintained" category). The numbers below each category show the number of the simulations out of the 200 ran that fall into that category. Y-axis values for each metric are in the units specified by that metric's label. Distributions are shown for each of the values. Since the Dempster-Shafer-informed driver is choosing between two profile options, it was expected that there would be many cases in which the comparison results in a zero difference, which skews the distribution. Beyond the zero difference comparison, the wear distribution shows a consistent advantage over the maximum deceleration profile, and the speed shows some advantage as well (i.e. Dempster-Shafer correctly recommended slowing down early which lead to higher speeds when the light changed). In most cases, while the Dempster-Shafer driver was farther from the intersection when the light changed, that was primarily five meters or less, which is an acceptably small difference.

when both vehicles do stop, the DS-informed driver waits longer at the light, which is again expected. Thus, the trade-off in this comparison is between wear and distance when the light changes: the DS-informed driver increases wear but is closer to the light each

time the light changes while ending at approximately the same speed as the variable rate deceleration driver, thus accumulating an advantage in distance at each intersection. With respect to the distribution, while there were again many cases with zero difference, this shows up most noticeably in the speed difference metric, showing that this choice clearly does not provide much advantage of speed over the variable rate deceleration. However, there is a consistently clear advantage in distance with the DS-informed driver ending up closer to the intersection while at the same speed when the light changes.

The third comparison is between the DS-informed driver and a coin-toss (50/50) choice between max deceleration and variable rate deceleration profile options. If the DS-informed analysis provides no consistent ability to correctly discern the time until the light changes to green, then it is reasonable to assume that it would not do better overall than a coin toss. Instead, the DS-informed driver clearly has lower wear in almost all cases, typically ends at similar speeds to the coin toss driver, and has a shorter wait at the intersection when both vehicles stop. However, the DS-informed driver is always further from the intersection when the light changes while the DS-informed driver is still moving. Comparing the mean with the range of values, though, shows that the distances are usually similar with a capacity for the DS-informed driver to be significantly farther from the intersection than the coin-toss driver. Interestingly, this comparison is very similar to the max deceleration comparison in Figure 4.7, although with less spread in the values. As with the maximum deceleration comparison, this comparison shows that most values are similar, with the wear and time to light change metrics being in favor of the DS-informed driver. With respect to the distributions, while there were again many cases with zero difference, each of the distributions other than distance shows a clear bias in favor of the DS-informed driver with higher speeds, lower wear, and a shorter time until the light changed to green. Interestingly, there are more zero distance difference cases than zero speed difference cases suggesting that while the two profiles more often ended up at the same distance from the light, the DS-informed driver had a speed advantage in more of those cases.

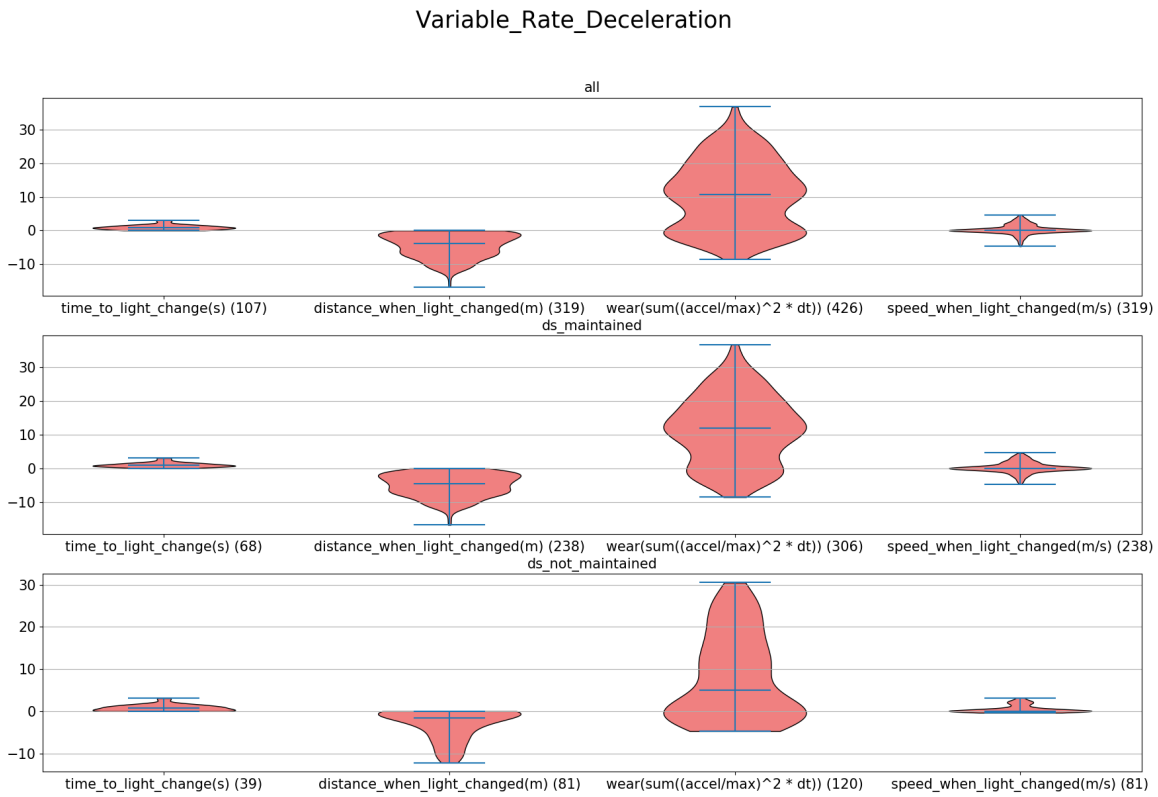

Figure 4.8: Comparison between the Dempster-Shafer-informed driver and a driver always following variable rate deceleration profile. Distributions are obtained by subtracting the results for the baseline profile driver from the DS-informed driver results for each simulation run. The three plots are All (include all data), "ds_maintained" (includes data in which the Dempster-Shafer evaluation told the driver to stay at speed at least 25% of the decisions), and "ds_not_maintained" (includes data not included in the "ds_maintained" category). The numbers below each category show the number of the simulations out of the total ran that fall into that category. Y-axis values for each metric are in the units specified by that metric label. Distributions are shown for each of the values. There is no clear advantage in speed between the driver and the variable rate driver when the light changes. However, the Dempster-Shafer-informed driver is consistently closer to the intersection when the light changes, leading to an overall position advantage. Based on the wear distribution, the trade-off is between wear and position advantage for this baseline comparison.

The final comparison is between a DS-informed driver and a Bayesian driver both choosing between the variable rate deceleration and the max deceleration. While both evaluations are dealing with uncertainty, the methods of entry and evaluation are different. In Bayesian belief networks (BBNs), without extensions for soft or virtual evidence [105], observations are entered with certainty (e.g. the state of the cross traffic is seen to be None, Light, or

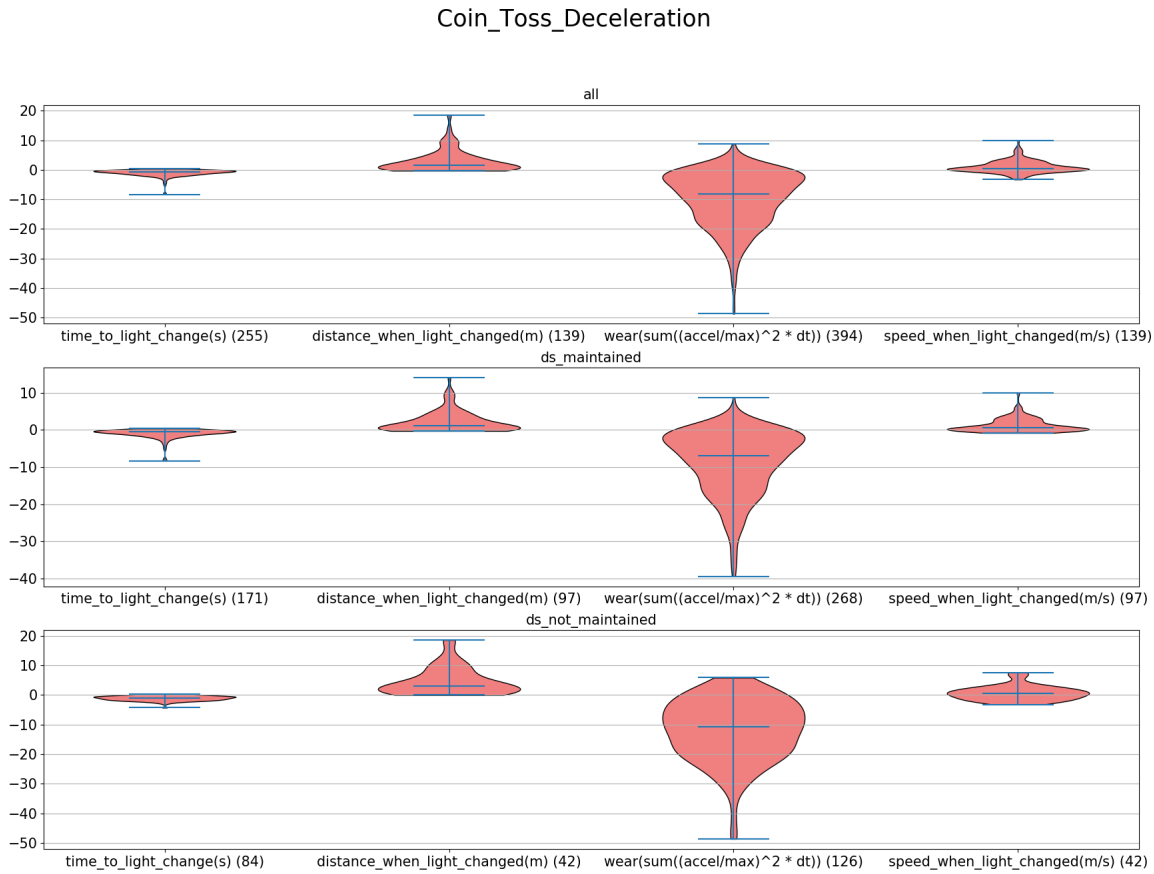

Figure 4.9: Comparison between the Dempster-Shafer-informed driver and a driver using a coin toss on each decision to choose between the max deceleration profile and the variable rate deceleration profile. Distributions are obtained by subtracting the results for the baseline profile driver from the DS-informed driver results for each simulation run. The three plots are All (include all data), "ds_maintained" (includes data in which the Dempster-Shafer evaluation told the driver to stay at speed at least 25% of the decisions), and "ds_not_maintained" (includes data not included in the "ds_maintained" category). The numbers below each category show the number of the simulations out of the 200 ran that fall into that category. Y-axis values for each metric are in the units specified by that metric label. Distributions are shown for each of the values. Other than in the distance metric, which shows a slight advantage to the coin toss driver, the other metrics show a clear advantage to the Dempster-Shafer-informed driver. Moreover, there are fewer zero difference speed cases than zero difference distance cases, suggesting that it was more likely for the two drivers to end up at the same distance from the intersection but with the Dempster-Shafer driver at a higher speed.

Normal). Conversely, as discussed in Section 4.2, DS theory by default enters evidence as uncertain, which allows concepts such as visibility to easily be mapped into an uncertain observation. Finally, note that the BBN conditional values were entered by a subject matter

expert based on the design of the simulation. A learning mechanism was not used for the BBN. The results of these comparisons are shown in Figure 4.10. While this comparison shows a Pareto frontier (i.e. no clear winner in all metrics), an equal weighting of metrics shows the DS-informed driver as the winner since the DS-informed driver performs better in three of the four metrics.

4.3.1 Window Size Effects

Real-time decision analysis in a constantly-changing scenario requires an appropriately-designed analysis window. If the window is too long, then the decision does not change rapidly enough to enable useful decisions. If the window is too short, the decisions either tend to change too rapidly or insufficient information is gathered to make a reasonable choice. A nice result of using DS analysis is that higher uncertainty due to a narrow window results in a consistent decision instead of rapidly changing decisions. Consider the results of a single simulation run corresponding to the traffic light scenario discussed in this chapter. Figure 4.11 shows the data analysis for the run. Due to the chosen window of five data points at 0.2 second intervals, the changes in analysis are relatively smooth, changing fully over a one second interval. Likewise, the decision graph clearly shows the change from believing there will be a long time until the light changes to green to believing there will be a short time until the light changes to green. As can be seen in the lower evidence graph, there is little unknown belief mass in the evidence, although there is significant ambiguity. This data is translated into decision analysis through the use of Table 4.1, which is shown in Figure 4.12. Before analyzing the lack of unknown evidence, it is useful to contrast this analysis against a smaller window, the corresponding graphs of which are shown in Figures 4.13 and 4.14. Note that the second set of graphs are for a different simulation run since the observations are generated from a normal distribution, making exact replication difficult. The evidence input in Figure 4.13 clearly shows more unknown than in Figure 4.11. The answer to this difference lies in the graph and prior combinations. Recall that

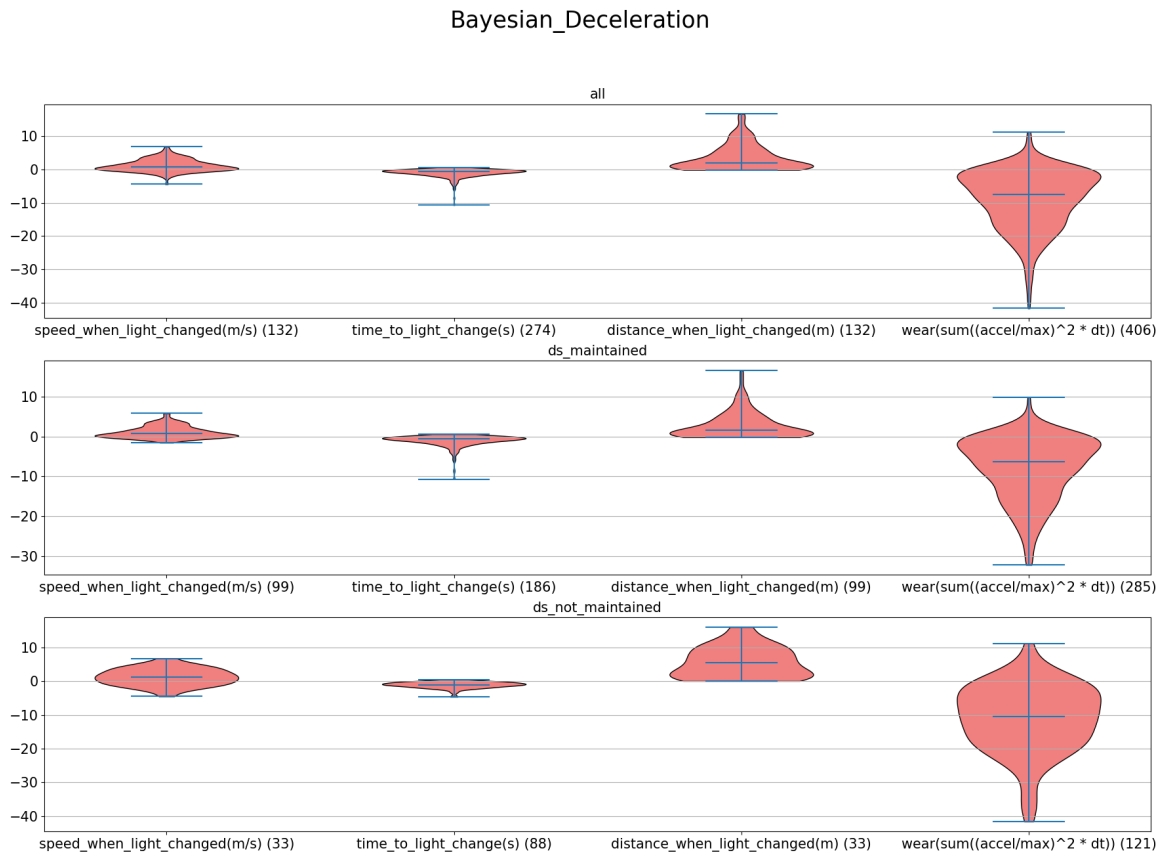

Figure 4.10: Comparison between the Dempster-Shafer-informed driver and a driver using a Bayesian evaluation to choose between the max deceleration profile and the variable rate deceleration profile. Distributions are obtained by subtracting the results for the baseline profile driver from the Dempster-Shafer-informed driver results for each simulation run. The three plots are All (include all data), "ds_maintained" (includes data in which the Dempster-Shafer evaluation told the driver to stay at speed at least 25% of the decisions), and "ds_not_maintained" (includes data not included in the "ds_maintained" category). The numbers below each category show the number of the simulations out of the 200 ran that fall into that category. Y-axis values for each metric are in the units specified by that metric label. Distributions are shown for each of the values. The distance metric shows an advantage to the Bayesian driver. The other metrics show a clear advantage to the Dempster-Shafer-informed driver. This comparison does show a Pareto frontier in that neither system is a clear winner in all metrics. However, since the Dempster-Shafer driver performs better in three of the four metrics, an equal weighting of metrics shows that the Dempster-Shafer driver performs better overall.

the "Time to Green" node is not observed directly during the evaluation stage of this scenario. Evidence inputs to other nodes, such as the "Traffic Movement" node, are combined at those nodes, the results of which are then provided through the graph to the "Time to

Green" node as evidence which is combined. This method results in decisions becoming more concrete as they pass through the graph, assuming there is sufficient evidence at each node for a reasonably concrete decision. The smaller window in Figure 4.13 caused the decisions at each node to remain less concrete due to scarce evidence, resulting in higher unknown values in the evidence inputs to the "Time to Green" node. Given those effects, a one-second window with five observations during that window resulted in a reasonable output for decision-making. Looking at Figures 4.12 and 4.14, it is also clear to see that the combination method and window size affect how the decision criteria must be stated. The original criteria in Table 4.1 was more appropriate for the smaller window size, given the higher unknown values. With the larger window size, the updated criteria provided a more accurate decision.

4.3.2 Network Learning Method Comparison

Multiple learning methods were evaluated as part of the traffic scenario. Since the general relationships between the state of the cross-traffic light and the time until the driver's light changes to green are known (red cross-traffic light typically means a short time until the light changes to green, yellow cross-traffic light means a medium time until the light changes to green, and a green cross-traffic light means a long time until the light changes to green), the learned relationships captured in the transitions can be evaluated. The first learning method, which updates the transition at each evidence input and retains the entire evidence history via Murphy's rule, resulted in the transition shown in Figure 4.15. While the weights stabilized quickly, the important note is that the weights are not representative of the expectations for this scenario. For example, the "(Green, Yellow, Red)" θ, which is the complete set (i.e. the unknown) maps strongly to the "Long" θ in the "Time to Green" node. Even the "Red" θ has a fairly strong mapping to the "Long" θ, which is unexpected.

Contrast Figure 4.15 with Figure 4.16. There are only four updates for this learning method

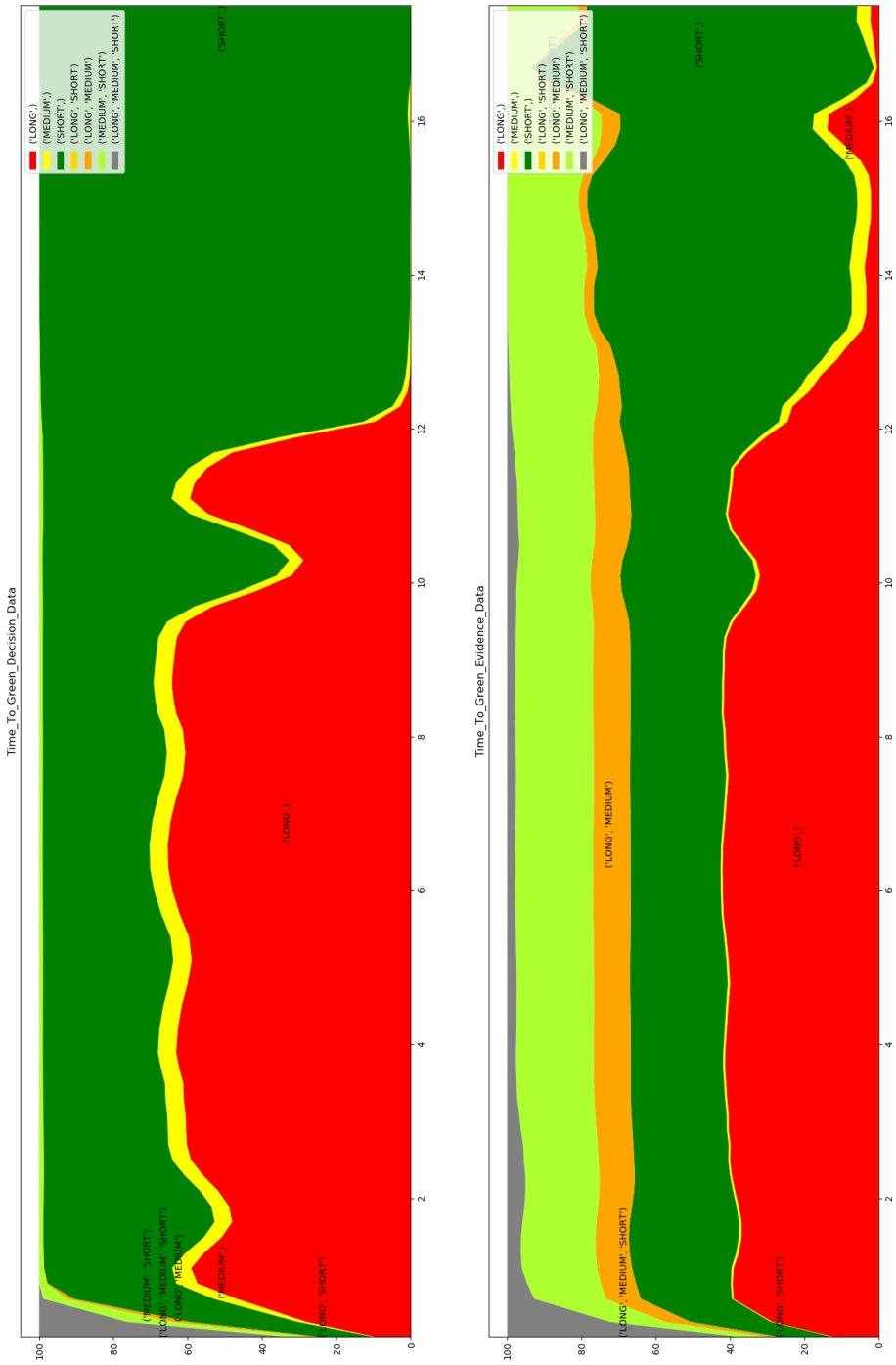

Figure 4.11: The Dempster-Shafer data analysis for a single run approaching the red light as a function of time in seconds. The upper graph shows the combined data using the Zhang combination method [18] over a window of five observations taken at 0.2 second intervals. The lower graph shows the evidence input at the "Time to Green" node at each time update. Due to the number of observations, the evidence input to the "Time to Green" node is smooth, with limited unknown belief.

since there was an initialization and three episodes, one for each traffic light state. However, it can easily be seen that the weights more closely align with expectations in which the "Green" θ strongly maps to the "Long" θ and to a much lesser extent the "Short" θ, the "Red" θ strongly maps to the "Short" θ, and the "Yellow" θ maps to the "Short" and "(Medium, Short)" θs, with a lesser mapping to the "(Medium, Long)" θ. Furthermore, unknown maps to unknown, which is required for a reasonable mapping.

Finally, Figure 4.17 shows episodic learning without direct evidence of the cross-traffic light. The principle effect of this change is that the node distribution for the cross-traffic light was being learned simultaneously and therefore updating as well. Consequently, the shifting weights between episodes affect a much larger portion of the transition than in Figure 4.16. This result corresponds to a "loss" of information since the simultaneous updates to the node prevents a non-overlapping transition update.

4.4 Conclusions

Overall, the DS-informed driver performed better than the four baselines chosen for evaluation. While the improvements were not always significant and represented a Pareto frontier (i.e. no clear winner in all metrics), they demonstrated that the DS-based decisions led to improvements in the driver response in this scenario. Based on the maximum difference in metrics between the deceleration profiles, it is reasonable to state that the DS-informed driver showed visible, consistent improvement over the baseline deceleration profiles. Further, the learning methods compared in Section 4.3.2 show a clear improvement in ability to capture the nuances of the relationships between node distributions, enabling the decision-making used in this scenario.

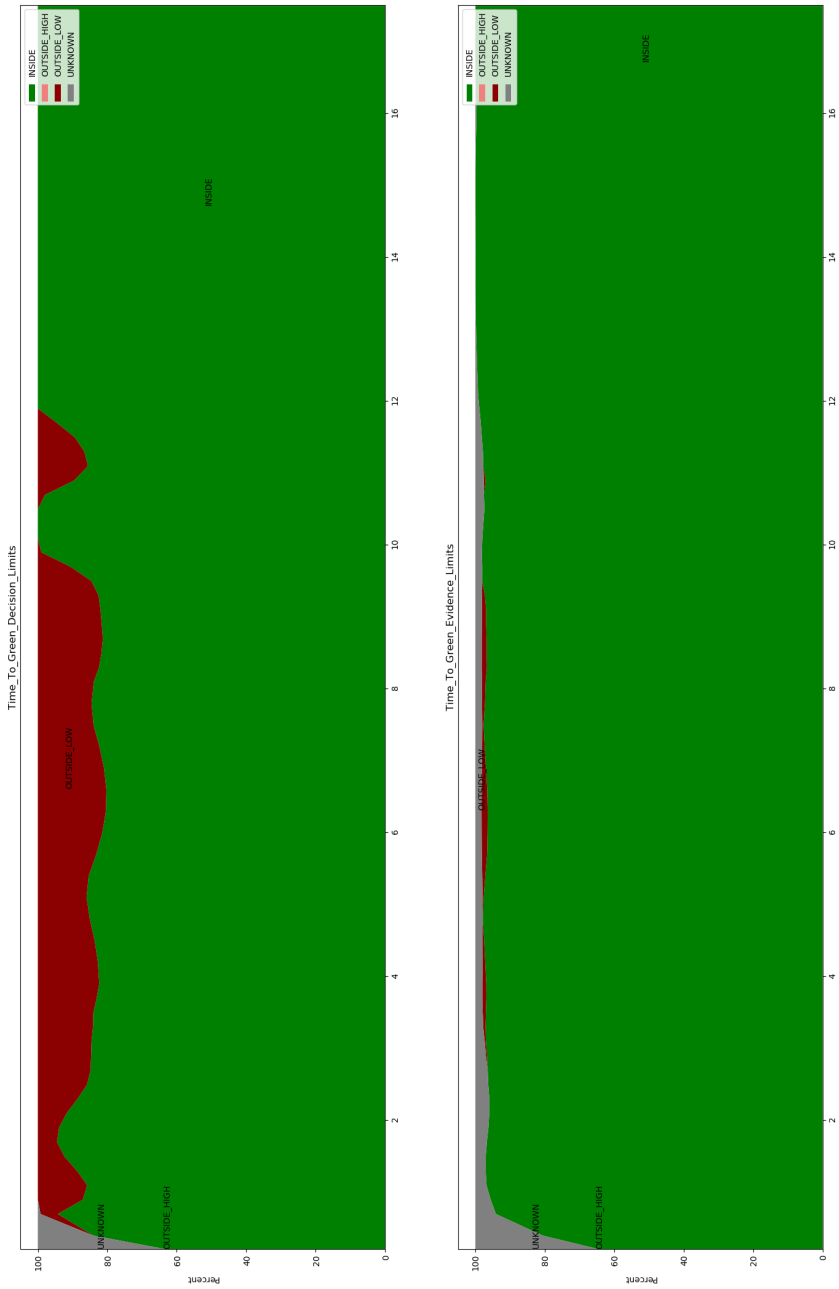

Figure 4.12: The Dempster-Shafer data limits analysis for a single run approaching the red light as a function of time in seconds. This analysis uses the data from 4.11 and the updated decision criteria from Table 4.1 to make the choice of whether to follow the maximum deceleration profile or the variable rate deceleration profile. The "Inside" label includes data that meets the decision criteria. All other data is "Outside". Any "Outside" data that is due to the complete set is shown as "Unknown". This graph shows that initially the decision criteria is close, but unmet since there is data outside the criteria (in red). At approximately 12s, the data meets the decision criteria (the full graph is green), allowing the decision-maker to proceed. As a result of the smoothing shown in Figure 4.11, the "unknown" data compared against the decision criteria is nearly non-existent, but the change in decision is smooth.

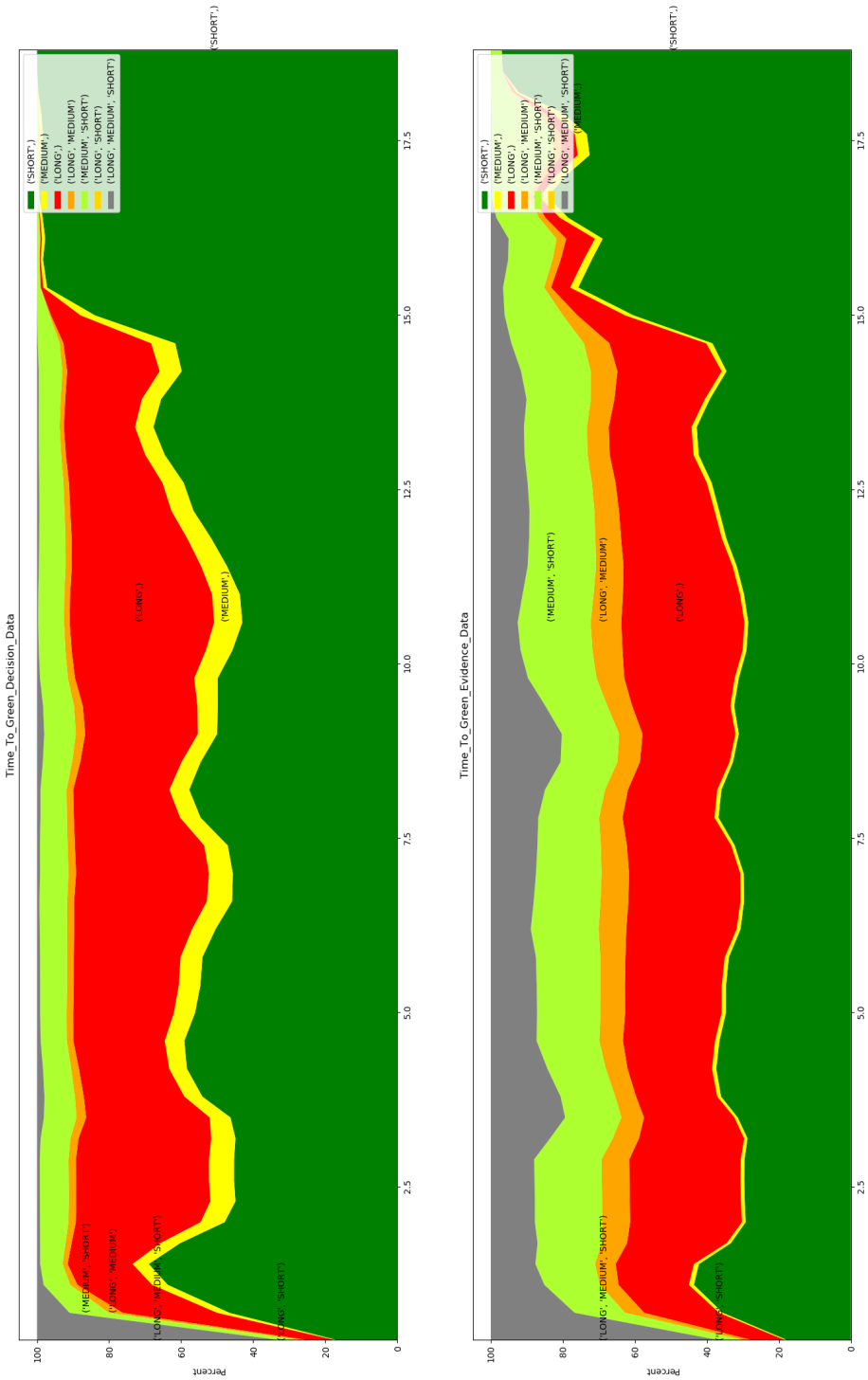

Figure 4.13: The Dempster-Shafer data analysis for a single run approaching the red light as a function of time in seconds. The upper graph shows the combined data using the Zhang combination method [18] over a window of three observations taken at 0.2 second intervals. The lower graph shows the evidence input at the "Time to Green" node at each step. As a result of the smaller observation window, there is a larger component of "unknown" evidence, and the shifts in evidence are less smooth.

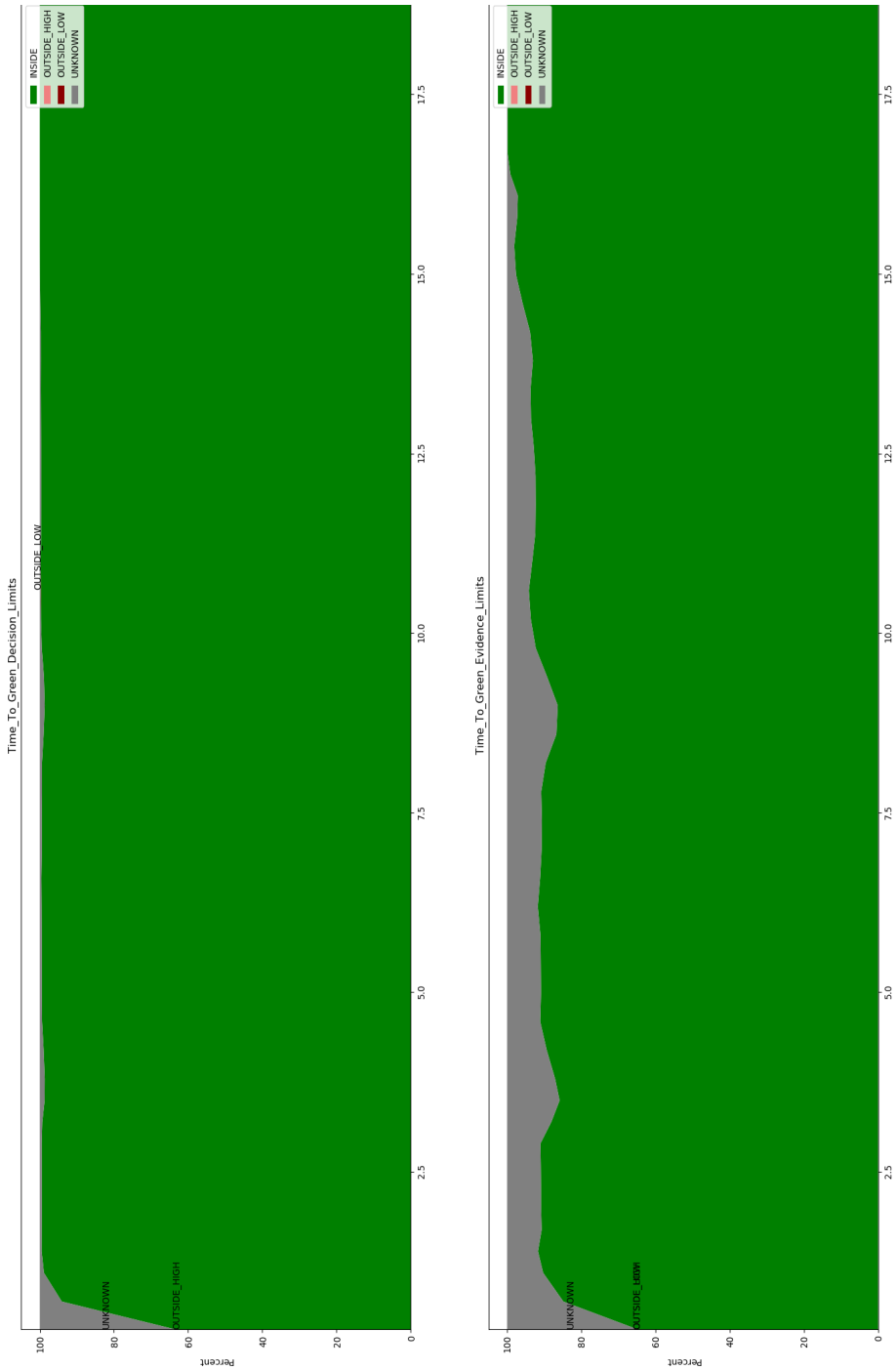

Figure 4.14: The Dempster-Shafer data limits analysis for a single run approaching the red light as a function of time in seconds. This uses the data from 4.13 and the updated decision criteria from Table 4.1 to make the choice. The "Inside" label includes data that meets the decision criteria. All other data is "Outside". Any "Outside" data that is due to the complete set is shown as "Unknown". As a result of the smaller observation window and the resulting evidence in Figure 4.13, there is a noticeable "unknown" component of the data compared against the decision criteria in the evidence, but the combined data quickly eliminates this unknown, resulting in a potentially premature decision.

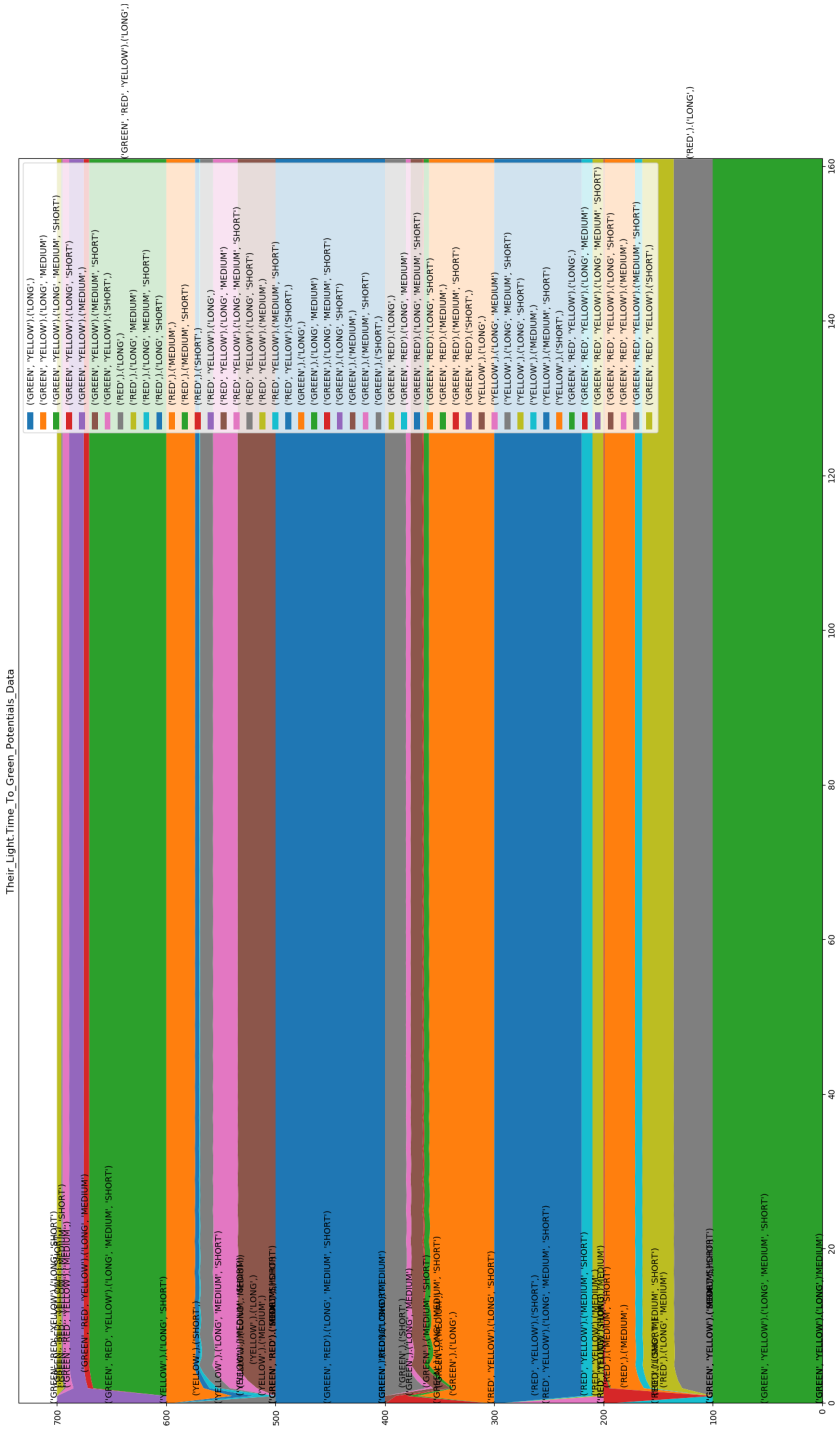

Figure 4.15: The transition between the cross-traffic light ("Their Light") node and the "Time to Green" node, showing the progression as the values were learned during the training phase. The x-axis shows each update as a step input. This learning method retained all evidence through Murphy's rule, included the state of the cross-traffic light as evidence, and did not incorporate episodic learning. As can be seen, the weights quickly stabilized and are not representative of the expected weights given the traffic scenario described.

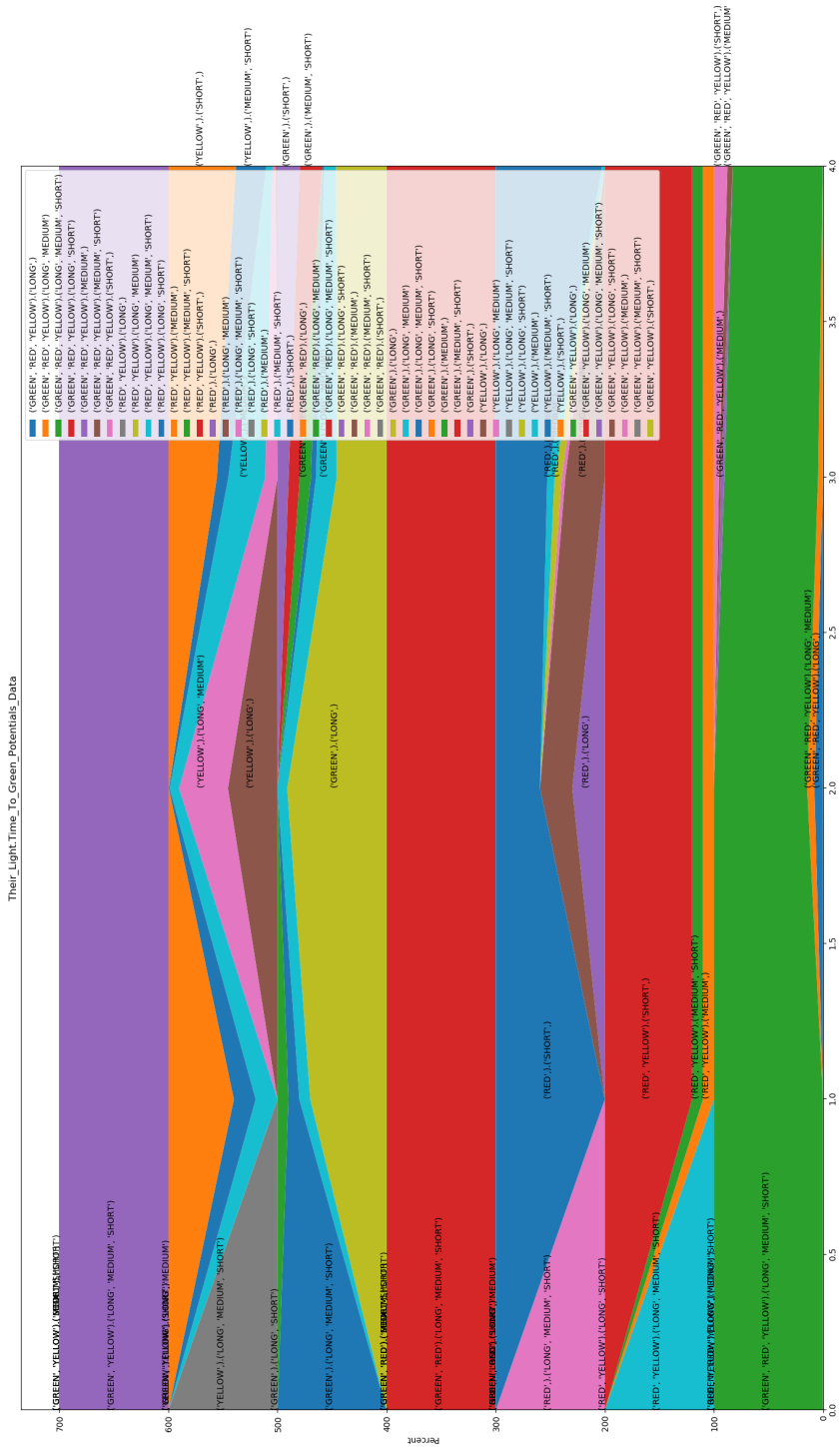

Figure 4.16: The transition between the cross-traffic light ("Their Light") node and the "Time to Green" node, showing the progression as the values were learned during the training phase. This learning method uses episodes comprised of the light states of green, yellow, and red. Further, this learning method includes evidence of the cross-traffic state. As can be seen, the weights better represent the expected weights for the scenario. The x-axis represents each episode as it was added to the network.

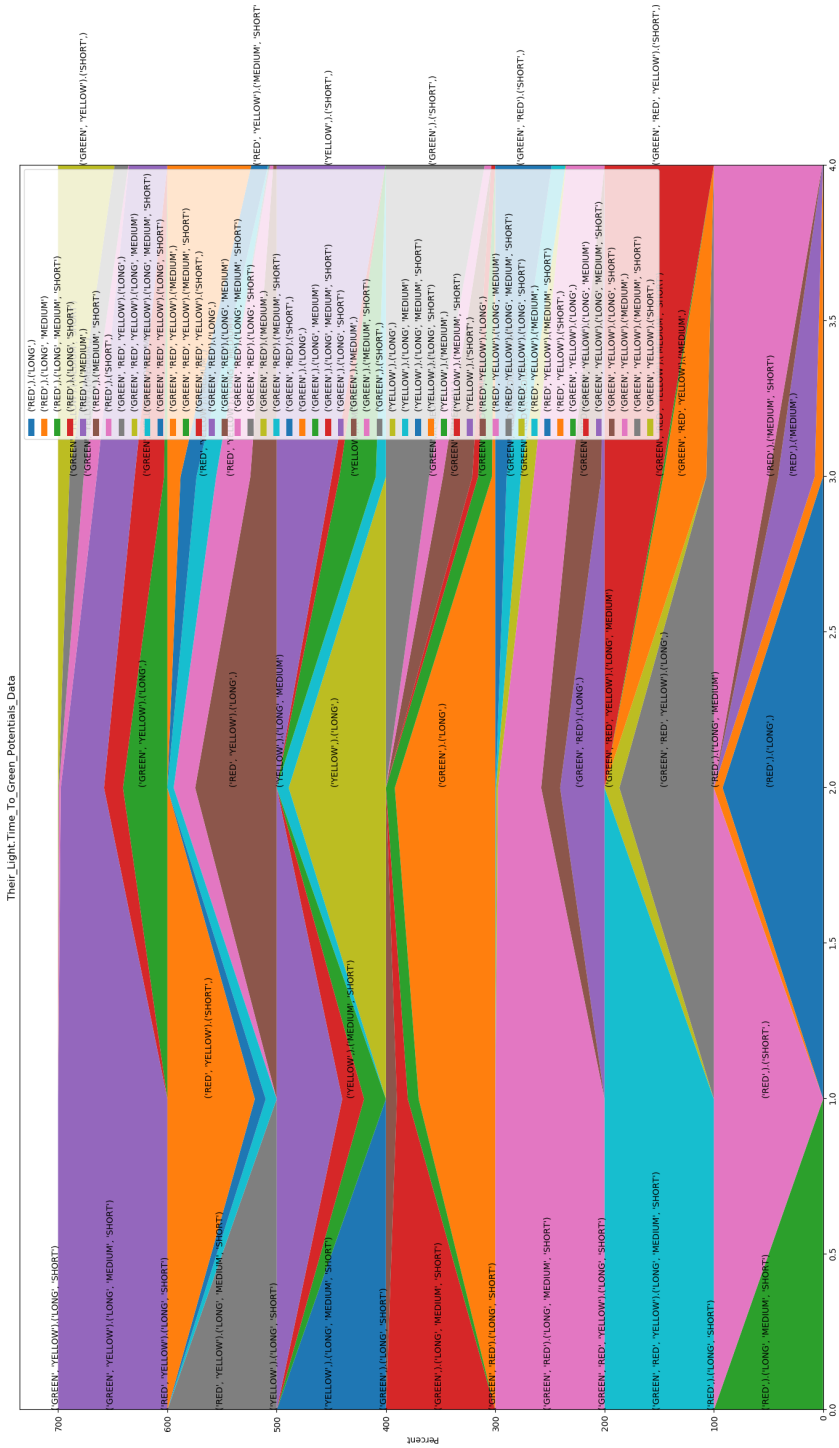

Figure 4.17: The transition between the cross-traffic light ("Their Light") node and the "Time to Green" node, showing the progression as the values were learned during the training phase. This learning method uses episodes comprised of the light states of green, yellow, and red. As can be seen, most of the stronger weights maps to short. Note that the mapping changes much more aggressively between observations than in Figure 4.16 suggesting that the node distribution was changing as well. The x-axis represents each episode as it was added to the network.

CHAPTER 5

UAS APPLICATION

The primary intent of this research, as discussed in Chapters 1 and 2, is to provide a risk analysis mechanism for systematically understanding the risks associated with UAS operations both *a-priori* and in real-time during operations. Chapters 3 and 4 detail the updates to the theory and implementation of the Dempster-Shafer (DS) analysis framework as well as provide an example of its application to driving. This chapter focuses on applying the DS updates to a UAS scenario — a hovering multirotor making real-time decisions on whether to land, and, if so, in which area to land. The scenario is shown in Figure 5.1.

Previous research in real-time health assessment of UAS [23] developed a system that provides good, warning, and failure indications of various UAS subsystems, which results in a comprehensive health diagnostic for the UAS. Three known limitations of this system are as follows:

- All subsystems must report a common basis of health diagnostic (e.g. good, warning, or failure).
- The comprehensive health diagnostic selects the worst case subsystem health diagnostic, which ensures a worst-case response only and does not provide a full understanding of the risks faced by the system.
- Configurable delays help to handle conditions causing rapid changes between instantaneous health diagnostic states (e.g. switching rapidly between good and warning), but these delays deal with this problem by introducing hysteresis mechanisms without capturing the underlying distribution between the states.

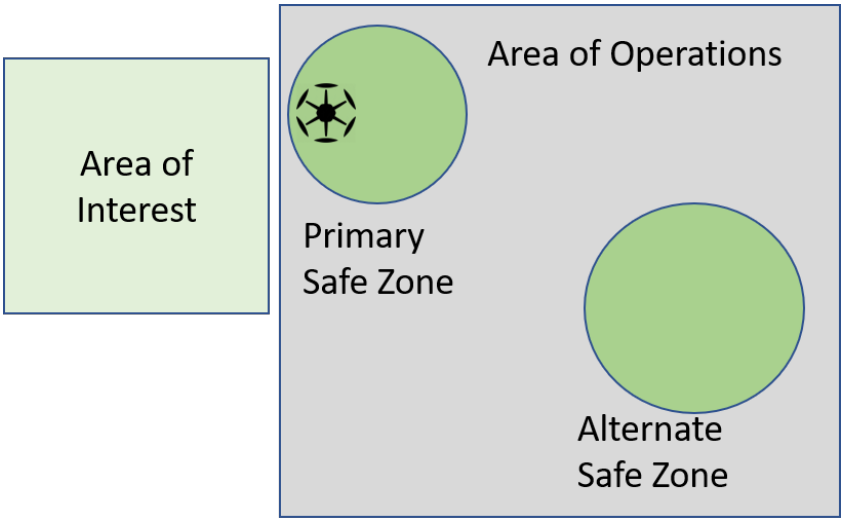

Figure 5.1: The layout for the UAS news multirotor scenario. An area of operations, which limits the risk of the UAS flight to lives not involved in the operation, is defined and shown. The goal of the news multirotor is to maintain the best visual coverage of the area of interest while maintaining an ability to land if issues arise, in order to keep the operation risk manageable. For this operation, two safe zones are specified as areas in which the multirotor could land without risk to lives. Additionally, the scenario assumes that some monitoring method for these safe zones are available, which could be as simple as an operator actively monitoring the zones and notifying the UAS if the zones are becoming unsafe for landing. Since the goal of the UAS is to maintain visual coverage of the area of interest, the multirotor hovers over the primary safe zone, but will move to the secondary safe zone if the primary zone is compromised. Additionally, the UAS will land if the risk becomes too high. This scenario encompasses many facets of UAS risk analysis including a mechanism to assess risk, multiple options/hypotheses, and decision criteria associated with the risk analysis.

Integrating this real-time health analysis system with a DS risk assessment overcomes these limitations by translating reported health diagnostics to a common basis through transitions, providing a full distribution including unknown and ambiguous components of the risk assessment. Further, the DS analysis easily incorporates additional information external to the UAS. Figure 5.2 shows a DS network that can be used for UAS real-time risk analysis.

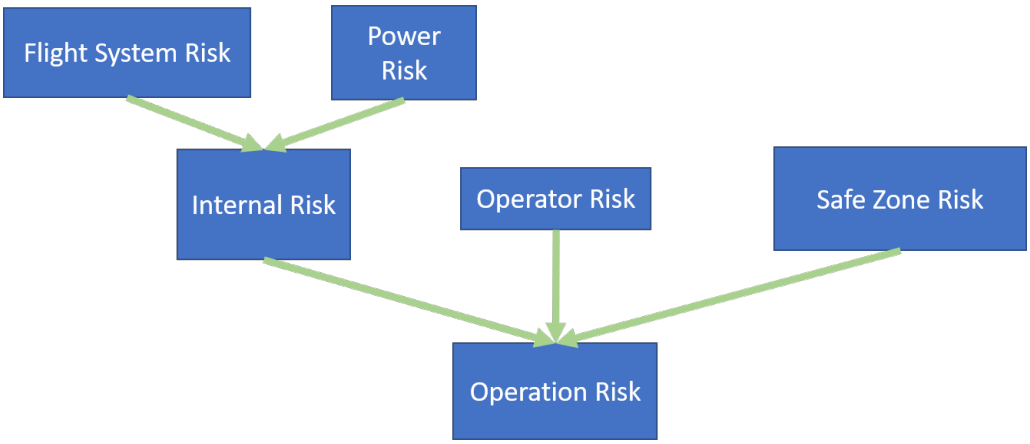

Figure 5.2: Dempster-Shafer network for UAS risk analysis. Each node includes three θs or individual options being evaluated: low risk, medium risk, and high risk. This network is more appropriate for a small, lower-cost UAS that will not respond differently to risks in each internal subsystem. Power and flight systems are still separated since power system warnings and failures are more common issues for multiirotors and have pre-planned responses. Likewise, the operator is a separate node since the capabilities of the operator (whether Part 107 [24] certified, etc.) play a strong role in the overall operation risk. Multiple risks can be assessed for the environment including weather, terrain, crowds, etc. Assuming that a Part 107 operator is correctly following rules and flying in appropriate weather for the UAS, the environment risk analysis is simplified to focus on safe zones, which are known, monitored landing zones for the UAS. This network includes multiple hypotheses for the safe zones, which are not depicted in this figure.

5.1 Decision Criteria Design

The UAS DS network includes two decisions. The first is the operation risk decision that determines whether the UAS should land or continue the operation. The second determines which safety zone should be used by the multi-rotor. If both safety zone hypotheses have the same result when compared to the decision criteria, then the primary hypothesis is used. Otherwise, the better hypothesis is chosen. This criteria is set more restrictively than the operation risk criteria such that the vehicle can pre-position itself before making the decision whether to continue the mission.

Table 5.1 shows the decision criteria values based on the decision types defined in Figure 4.6. Since this network requires one transition for most inputs to provide evidence to

the operation risk node, the decision analysis in the operation risk node will be less stark than in the traffic light scenario (Section 4.3), thus enabling a greater difference between belief and plausibility bounds.

Table 5.1: The numerical decision criteria for the UAS Dempster-Shafer network. For real-time flight system implementation, the decision criteria is reversed such that when the decision criteria is met, the system executes the contingency action. As such, these criteria are consistent with Figure 4.6. The Safety Zone Risk criteria are more restrictive than the operational risk, enabling the hypotheses and the chosen safety zone to be switched before deciding to end the mission, thus providing an opportunity to continue the mission with a lower risk safety zone.

Criteria Set	Limit Application	Low	Medium	High
Operation Risk	Belief	≤ 0.5		
	Plausibility	≤ 0.9	≥ 0.4	≥ 0.3
Safety Zone Risk	Belief	≤ 0.6		
	Plausibility	≤ 0.9	≥ 0.3	≥ 0.2

An important decision design criteria is the desired relationship between the probability of detection and the probability of false alarm — where the system is placed on the Receiver Operating Characteristic (ROC) curve [106]. The baseline system was designed to not produce false alarms (i.e. the probability of false alarm is zero unless a subsystem incorrectly provides a false alarm to the health monitoring system) [23]. The reason for this design decision is that the baseline system is typically flown on an experimental/research platform with a safety pilot as part of the flight test. It is assumed that the safety pilot can handle cases that the baseline system does not catch, and that false alarms could be detrimental to the flight test. The decision design criteria for the DS system under test was set to approximately match the baseline system such that a direct comparison could be made between the two systems, instead of relying on a Pareto frontier for comparison. As such, it is likely that there will be no false positives during the evaluation in Section 5.4, although this is still included as an evaluation criteria for completeness as stated in Section 5.2.

Where the DS system is placed on the ROC curve is a function primarily of the decision cri-

teria (Table 5.1), the reaction time required, the reporting of the individual subsystems, and the evidence translation from the instantaneous reports of the individual subsystems (Table 5.2). If it is acceptable for individual subsystems to report degradation with some probability (e.g. GPS may occasionally report lost lock if it recovers within a given time frame) without the system taking contingency action, then that limit needs to be handled when defining the decision criteria. In the case of the UAS platform on which this system will be demonstrated (see Section 5.5), it was assumed that any internal subsystem degradation could cause a safety issue, and the system should take contingency action. However, for the safety zone evaluation, some degradation was allowed since the DS evaluation needed to test both hypotheses before determining whether to end or continue the mission. Figures 5.11 and 5.12 show that a small overall change to mission risk was allowed without ending the mission. Since the overall mission risk is used to evaluate whether to end the mission for these tests, the required reaction time was fast (1-2sec for a complete failure reported by a subsystem), resulting in limited degradation allowed. For a more complex network, such as described in Section 5.7, multiple decision criteria on different nodes can be defined to enable different reactions and different points on the ROC curve [106], as long as the least restrictive decision criteria on the overall mission risk still meets required risk limits.

5.2 Evaluation Metrics

Evaluation of this real-time risk analysis method was performed in comparison against the baseline real-time health assessment architecture implementation in GUST [70] [23]. For this comparison, the following metrics were chosen:

- Response time: the time in seconds between the first notification of a degradation in the system (e.g. an instantaneous health status changes from good to warning) and when the system takes contingency action in response to the change

- False positives: the number of times the system takes contingency action when there is no degradation in the system
- False negatives: the number of times the system does not take contingency action when there is a degradation of the system
- Processability: whether the system is capable of processing the full autopilot update including the DS network in the 100Hz update loop

For each system (the baseline and the DS network — i.e. the system under test), trade-offs between the first three metrics are handled through configuration parameters. These parameters were chosen to provide a similar balance between each of the first three metrics before comparisons were made between the systems. The final metric is a requirement for running on small UAS, which is previously demonstrated on the baseline system [23] and must be evaluated on the test system. Since the baseline system did not include the concept of safe zones and multiple hypotheses, that capability is evaluated separately by testing whether the UAS can choose the less risky safe zone (relative to the decision criteria defined for safe zone choice) and delay aborting the mission by lowering the risk through the safe zone choice. Further, the baseline system does not provide a way to include health or risk information external to the system, such as operator risk. This capability will also be tested separately by evaluating whether the UAS chooses the defined contingency action when risks from external systems become too high or whether the UAS continues the mission with no change.

5.3 Network Training

Training for the DS network was performed offline via simulation. While online training is feasible through this system, current methodology, as discussed in Section 3.4, uses the results of each operation to learn the transition potentials — the relationships — of

the network. Specifically, this method connects the long-term and real-time in-operation timelines through the transition potentials. Since DS combinations are used to combine the evidence per episode for the transition potentials, the captured relationships can change over time as the evidence suggests shifts. Offline training was chosen for this application due to the number of operations that were necessary to capture reasonable relationships for testing against the baseline system. To clarify, evaluating only a few test cases results in significantly higher levels of uncertainty mapping between nodes, which does not provide a similar response to the baseline system. Further, the approximate relationships were known *a-priori*, enabling rapid confirmation or disconfirmation through offline training.

Failures were modeled as Poisson Point Processes [107], which assumes stochastic independence between each failure occurrence. More accurate failure models are likely available on a manufacturer-by-manufacturer basis such as by Velos Rotors [108], which provides a maintenance schedule for their UAS parts. The simulation used for this research includes a replaceable model, and the Poisson Point Process was chosen since the flight demonstration platform, described in Section 5.5, is experimental with too few flight hours of data to provided a more accurate model. Offline training was performed by setting the probability of failure and warning (failsafe) per hour for each of the nodes without parents (flight systems, power systems, operator, and safety zone). The probabilities of failure and warning were set to zero and 50% each during training cases. While 50% failure probability is high, it gives a view into the relationships when components are at high risk of failure. Simulation lengths were randomly chosen between the minimum of 0.1 hrs and maximum of 30 minutes, assuming a commercial multirotor such as a DJI Mavic Pro [7]. 250 operations were run per learning case.

As with the traffic light scenario in Section 4.2, conversion functions from the subsystems' instantaneous reports of good, warning, or failure to evidence input for the network

has a substantial impact on the training and how quickly the transition potentials converge. Table 5.2 shows the chosen mapping from result to evidence input. Ambiguities and unknown masses have an impact on how the DS solutions converge, similar to the results in Chapter 4. Higher ambiguities and unknown results in slower, more consistent convergence, but there is a theoretical limit as detailed in Appendix A, since the specific thetas must be greater than the related ambiguities. Moreover, the ambiguities, while necessary for smoothly combining the evidence, also have real-world meaning. A successful operation does not mean that the risk was necessarily low. It merely means that the risk did not manifest itself into a failure or failsafe. Therefore, the (success, failsafe) ambiguities and the (success, failsafe, failure) unknown mass capture this possibility. Higher ambiguities and unknown masses highlight greater uncertainty between the operation results and the risks in the operation, allowing more operations to reduce the uncertainty through evidence combination.

Table 5.2: The mapping from results to maximum risk evidence inputs for each element of the power set. Actual evidence inputs for each operation are randomly selected up to the maximum values in the table, with any leftover mass being assigned to the complete set (unknown) to ensure the evidence masses always add up to 1.0. This partially random input simulates evidence from risk assessments performed after the flight operations for each specific operation. In practice, the high number of runs along with Murphy's combination rule [19] for averaging inputs results in a similar outcome to fixed values. The operation risk has a higher degree of uncertainty per outcome to simulate the uncertainty associated with the risk of the overall operation. Individual subsystem risks are more precise, simulating more detailed fault analysis on the subsystems.

	Operation Risk			Individual Risks		
Power Set	Success	Failsafe	Failure	Success	Failsafe	Failure
Low	0.4	0.0	0.0	0.7	0.0	0.0
Low, Medium	0.3	0.15	0.0	0.3	0.3	0.0
Medium	0.0	0.4	0.0	0.0	0.7	0.0
Medium, High	0.0	0.22	0.3	0.0	0.4	0.3
High	0.0	0.0	0.4	0.0	0.0	0.7
Low, Medium, High	1.0	1.0	1.0	1.0	1.0	1.0

A significant shift from the training for the traffic light scenario evaluated in Chapter 4 is

the need to capture overlapping training episodes for the transition updates. Training for the traffic light scenario, as detailed in Section 4.3.2, was greatly simplified due to the network primarily being based on the state of the cross-traffic light. While the cross-traffic density had an impact, most observations (cross-traffic speed, the time until the light changed to green, and cross-traffic walk signal) depended primarily on the cross-traffic light, thus allowing the episodes to be defined by the state of the cross-traffic light. Conversely, in Figure 5.2, it can be seen that all nodes affect the operation risk, creating a situation in which no one node correlates to all of the possible episodes. As for the traffic light scenario, to avoid implementing statistical heuristics to determine episodes, expert information was used to define the episodes. However, in this case, the results of each operation affected more than one episode. As with the traffic network in Figure 4.4, the impacts of interest per episode are defined by which transitions are affected through operation observations. Figure 5.3 shows the relationships between downstream nodes and nodes that are directly controlled in training episodes. For example, an episode of training that is running operations with a high likelihood of failure in the flight systems will teach the network about the transitions between the "Flight Systems Risk" node and the "Internal Risk" node and between the "Internal Risk" node and the "Operation Risk" node. However, the transition from the "Operator Risk" node to the "Operation Risk" node is not affected significantly, and including the effects of the episode on that transition is detrimental to the training of that transition.

A third and final restriction was added to training transitions of multi-parent nodes. In the two previous methods, even if there was only a significant change in the distribution of the "Power Risk" node (i.e. the "Flight Systems Risk" node had no effect on the change in risk of the "Internal Risk" node), the two transitions are still trained together. This is a direct result of the simplification in Section 3.3.3 in which the result of the parent nodes multiplied by the transition potentials are assumed to be the same when provided as evidence to

the combination method in the child node. This third method still uses that simplification, but then only trains the transition to which the episode applies. Figures E.2, E.4, and E.6 in Appendix E show the differences in training results between these three methods.

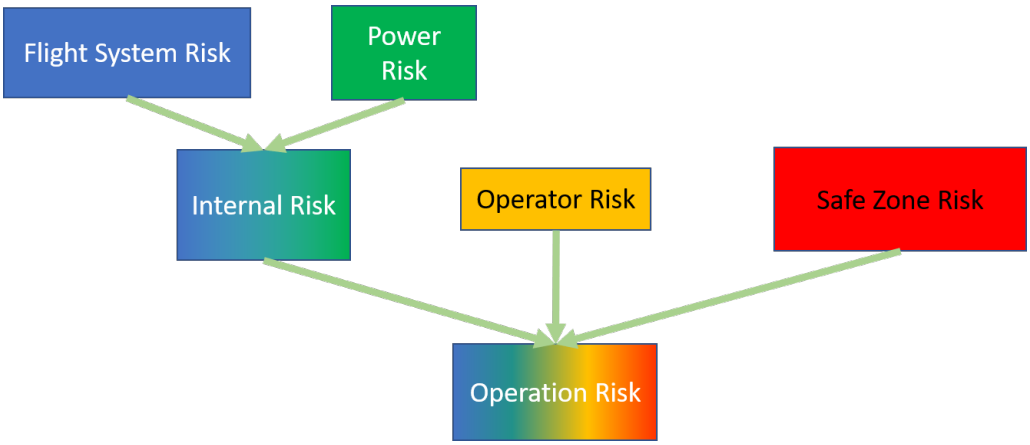

Figure 5.3: UAS risk analysis Dempster-Shafer network showing which nodes are primarily affected through changes to the evidence inputs. For example, changes to the Flight System Risk primarily will affect the Internal Risk and Operation Risk, leading to those three nodes being trained as part of the same episodes.

5.4 Test and Evaluation

Given the network and metrics defined previously for this scenario, multiple tests were created to evaluate the systems relative to the metrics in Section 5.2. 50 trials per test case were then conducted.

- Simple switch: The instantaneous health notification of a subsystem step changes from good to warning without switching back.
- Continuous switch: The instantaneous health notification of a subsystem continually changes on each update between good and warning.
- Stochastic switch: The instantaneous health notification of a subsystem changes between good and warning with some defined probability. A goal of this test is to determine whether there is a point at which the baseline and DS systems clearly

differ in their response.

Testing capabilities were built into the GUST flight control to enable both simulation and in-flight tests while minimizing risk to the UAS. Each subsystem in GUST for which there is health analysis provides an instantaneous health status of good, warning, or failure to the health subsystem, which uses those statuses to evaluate the overall status of the system [23]. A low risk method of testing the health system response while not degrading critical flight control systems is to inject a test system that reports an instantaneous good, warning, or failure status into the health subsystem. Test code was added to compute contingency response time based on the test subsystem injection and to check whether the health subsystem caused a contingency response without a degradation being injected by the test subsystem.

As with the network training in Section 5.3, subsystem instantaneous reports of good, warning, and failure had to be converted into evidence inputs for the DS network. This conversion is simpler than the training case because these are short term evidential views that are equal in weight and constantly updating. Thus, the simple mapping in Table 5.3 is used.

Table 5.3: The mapping of instantaneous subsystem health state to evidence input into the Dempster-Shafer network used by the health subsystem. Since each of the instantaneous health states are a form of evidence, a simple mapping is used to add ambiguity and unknown, allowing the Dempster-Shafer combination algorithms to work effectively and providing a slower, less stark response to changes in state.

		Instantaneous Status	
Power Set	Good	Warn	Fail
Low	0.6	0.0	0.0
Low, Medium	0.2	0.2	0.0
Medium	0.0	0.4	0.0
Medium, High	0.0	0.2	0.2
High	0.0	0.0	0.6
Low, Medium, High	0.2	0.2	0.2

Results are grouped to show the trends in the baseline system, the DS network system, and comparisons between the two. All test cases did not exhibit any frame overruns, meaning that both health analysis methods completed all executions within the 100Hz frame update rate in the simulator. The results of the baseline system is shown in Figure 5.4. The baseline system is a deterministic system with a one second delay before taking contingency action. The system is designed to ignore short, intermittent failures, but captures sustained or significant system degradations. However, as seen from Figure 5.4, the deterministic system is barely able to capture a highly degraded system. The inverse of the system would behave similarly: if the system is biased towards safety, then only slight degradations would result in the system taking contingency action.

The results from the health subsystem using the DS network are shown in Figure 5.5. The results act similar to a low pass filter in the sense that the network more slowly catches failures as sufficient percentages of the instantaneous health reports are "warning" or "failure" over a window of time. In this case, the update rate is 0.2 seconds, and the window is 10 sets of evidence, resulting in a two second window for determining failure conditions. For these evaluations, Zhang's combination rule [18] is used, since this provides nice properties for eliminating outliers and provides a faster response to substantial degradations. With these settings, this method captures failures nearly as quickly as the baseline system for simple failures and can capture most failures down to 25% probability of reporting an instantaneous warning or failure status, which is well beyond what the baseline system can capture.

The results in Figures 5.4 and 5.5 are compared directly in Figure 5.6 for the cases in which both methods can capture failures. While the baseline method responds faster for simple failures, the DS network clearly captures more failure cases and provides consistent responses to those failures. These two systems create a Pareto frontier. However, given that the DS network has a similar response time to the baseline system for simple failures

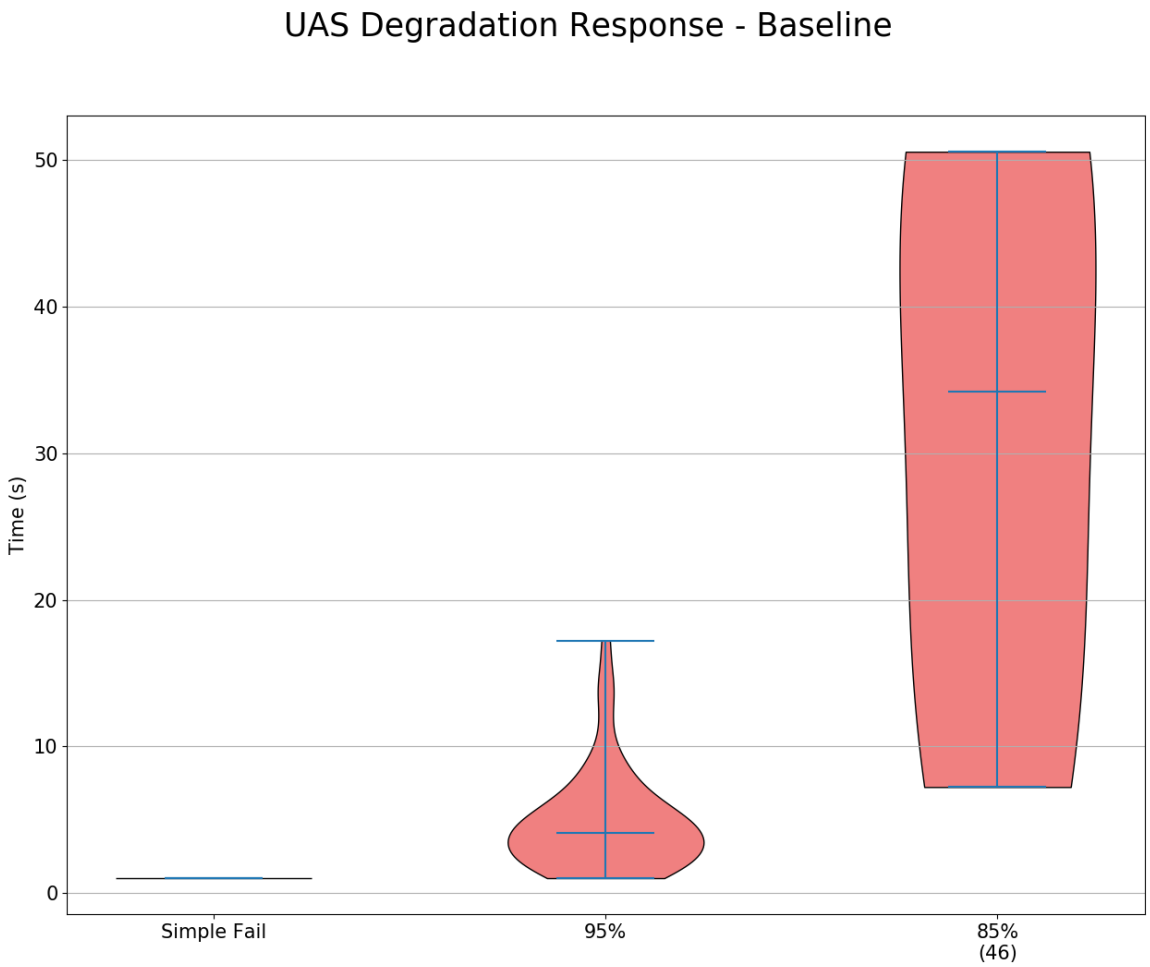

Figure 5.4: UAS baseline response to subsystem degradation injection into the health subsystem. The system is graded on response time. Numbers in parenthesis below the trials indicate the number of false negatives (uncaptured degradations) in the 50 trials. No false positives were captured. Note that this system is biased away from false positives to avoid safety maneuvers during flight tests. A safety pilot is assumed to be present during flight tests since this is an experimental aircraft. Simple failure is the same as 100% probability of instantaneously reporting failure (i.e. the system simply fails and continually reports a failure). Percentage failures are the probability that the subsystem will instantaneously report failure (i.e. the subsystem is degrading, but not fully failed). Lower percentages than 85% are not shown since no failures were captured at 80% or below.

and out-performs the baseline system for other failures, overall the DS network clearly performs better than the baseline system.

As stated in the metrics definition in Section 5.2, further capabilities are provided by the DS network method that are beyond the scope of the baseline system. These capabilities

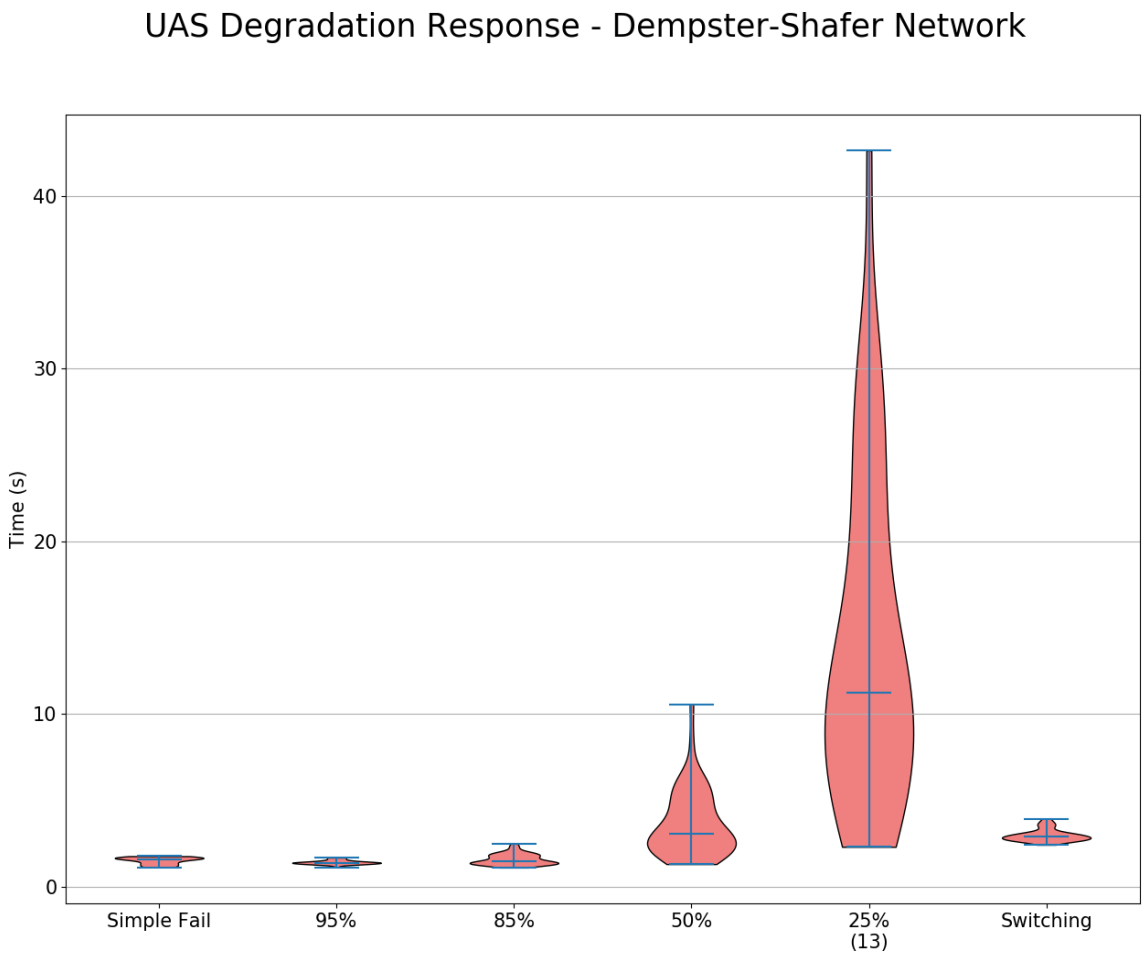

Figure 5.5: UAS Dempster-Shafer network response to subsystem degradation injection into the health subsystem. The system is graded on response time. Numbers in parenthesis below the trials indicate the number of false negatives (uncaptured degradations) in the 50 trials. No false positives were captured. In order to have comparable results to the baseline system, this health subsystem was also biased away from false positives, meaning that significant deviations from the "good" distribution were required to trigger a contingency action. Two noteworthy points arise from these results: (1) all contingency response times have a distribution — even the simple failure case — since the response is no longer deterministic. (2) this method captures down to 25%failure, albeit with some false negatives and significantly longer times to capture the failure. Moreover, this system gracefully degrades in the sense that the tail of the distribution extends consistently as the failure rate lowers. Test case meanings are the same as in Figure 5.4. The final test case — switching — is a case in which the subsystem alternates reporting good and failure on every update. This is a case that is impossible for the deterministic baseline to capture, but the Dempster-Shafer network captures this quickly.

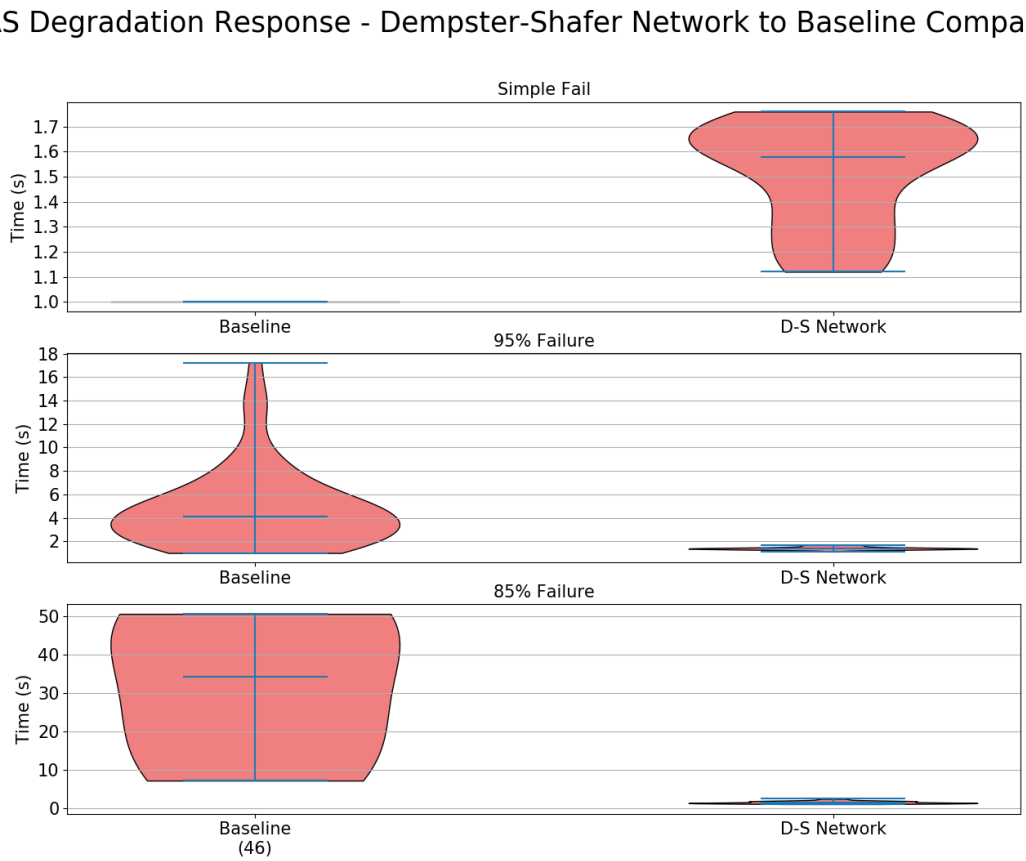

Figure 5.6: UAS health subsystem response comparison between the baseline method and the Dempster-Shafer network method. Only cases in which both methods can capture failures are shown. The baseline system clearly reacts faster for simple failures, but the Dempster-Shafer network method has a consistent, albeit slower, reaction for both the simple failure and lower failure reporting probabilities. The Dempster-Shafer network model clearly captures significantly more failure cases while not slowing the response time substantially.

are evaluated separately by determining if the system responds correctly to higher risk in the operator or safety zone analyses. Three test cases were run with 50 test runs per case. The first test case increased the medium risk evidence of the operator. The second case increased the medium risk evidence of the primary safety zone, leaving alone the secondary safety zone. The final test case increased the medium risk evidence of both safety zones together. These risk evidence inputs were pseudo-random. Limits were provided for each of the subsets within the risk evidence, and the evidence was pseudo-randomly chosen up to those limits. Any remaining mass was placed in the unknown set. This method simulates

evidence based on observations since these inputs were not driven by actual subsystem calculations, unlike the previous tests. As seen in Figure 5.7, degradations are captured successfully for both cases. The response time to the safety zone degradations is longer than the response to the operator degradation, which aligns with first checking whether the alternate hypothesis is a better choice. The cases in which only the primary safety zone had increased risk are not shown since, in these cases, the health subsystem correctly switched hypotheses and did not take any contingency action.

5.5 Demonstration

This research was grounded in the application of risk analysis for unmanned systems. Further, a key assumption was that this research could be applied to the full UAS ecosystem — small systems through large systems. To show that this assumption holds, a flight demonstration was performed using the vehicle in Figure 5.8 and shown in flight in Figure 5.9. This vehicle uses a Raspberry Pi 3B embedded computer with an Emlid Navio2 [109] autopilot sensor suite running the GUST [70] flight control system. This vehicle was previously used to demonstrate the GUST health monitoring architecture [23], providing a baseline for expanding to the DS network. For this flight demonstration, the trained network from Section 5.3 was loaded onto the UAS, mimicking the testing performed in Section 5.4. Once the UAS was airborne, two evidence injections were used from Section 5.4:

- Higher risk at the primary safety zone, which should cause the UAS to move to the secondary safety zone
- Higher operator risk, which should cause the UAS to land

The primary question of whether the DS network can run efficiently on the embedded flight control system is answered in Figure 5.10. Computations including the DS network stay

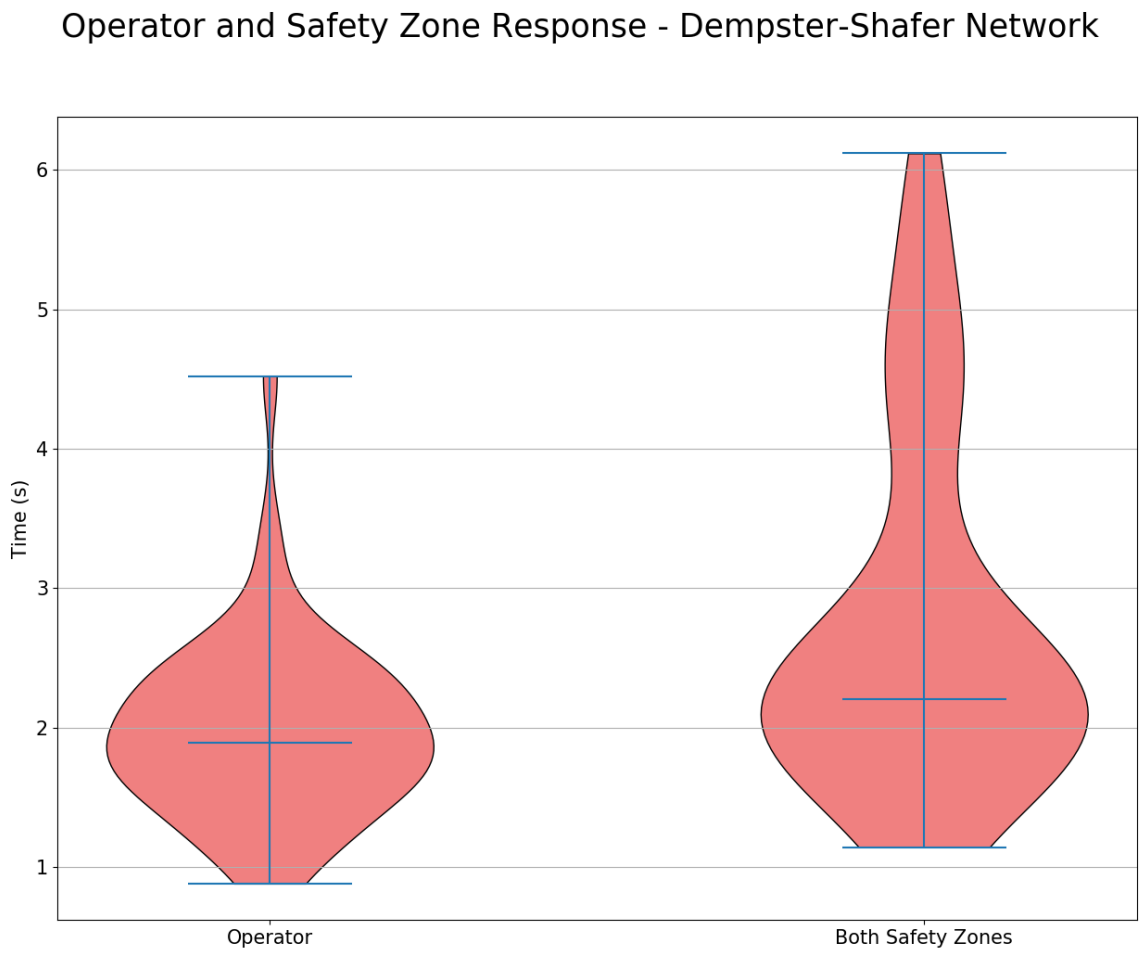

Figure 5.7: UAS Dempster-Shafer network health subsystem responses to increased operator and safety zone risks. Only two cases are shown — the operator risk increase and the dual safety zone risk increase. All risk increases were captured. The dual safety zone risk increase has a longer response time as the two hypotheses (the dual safety zones) are first considered to determine whether there is an alternate option before deciding to take contingency action. The single safety zone risk increase test is not shown since the UAS never took contingency action in this case. Instead, the UAS chose the secondary safety zone for landing when necessary.

within the time allotted for the 100Hz fixed frame update rate, demonstrating the DS-based system can run on small embedded computers.

The injected risk evidence and contingency responses for the two flight demonstrations are shown in Figures 5.11 and 5.12. Recall that the injected evidence is the same as in Section 5.4. As such, only the low risk belief value is shown in the figures, as this provides a surrogate for when the updated evidence is injected. In both demonstrations, the recorded data

Figure 5.8: UAS research platform used for the flight demonstration. Flight control and onboard computing is provided by a Raspberry Pi 3B embedded computer with an Emlid Navio autopilot sensor suite. UAS frame size is 400mm. A small platform and basic embedded flight computer was chosen to demonstrate applicability to the full range of UAS sizes, as larger platforms can carry more powerful computers.

starts with the UAS hovering over the primary safety zone. When updated evidence for the primary safety zone (SZ1) is injected, the operation risk shows a momentary change in risk analysis, but recovers quickly as the secondary safety zone hypothesis takes over. Concurrently, the UAS exits the hovering state and flies to the secondary safety zone, proceeding to hover there. When the higher risk operator evidence is injected, the overall operation risk increases, triggering the second flight plan, which is an immediate landing. Upon landing, the UAS returns to waiting for the next command. Both flight demonstrations show a consistent response to the inputs, and both flight demonstrations are consistent with the testing results seen in Section 5.4.

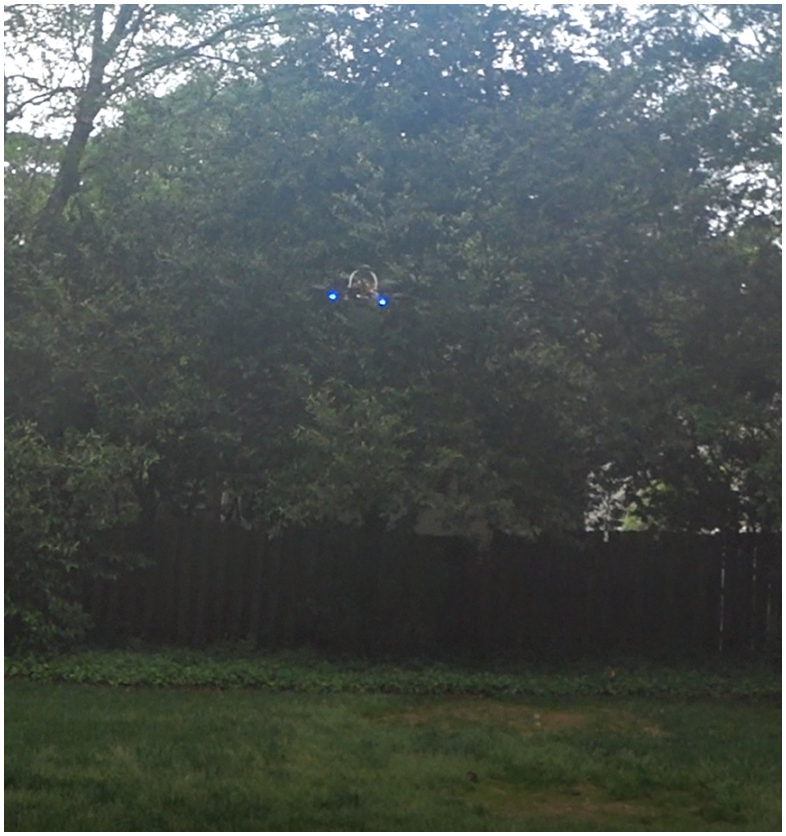

Figure 5.9: UAS research platform in flight during the flight demonstration. The flight demonstration was kept to a constrained area for personnel safety. All onboard health decisions were performed through the Dempster-Shafer risk analysis network.

5.6 Conclusions

The DS network significantly improved performance over the baseline system for health monitoring. Contingency response to full failures were similar to the baseline system, and the DS network detected intermittent failures far beyond what the baseline system could detect, with a graceful degradation in response time. Much of this improvement was realized by using a stochastic system, which would only require a single DS node (i.e. not require the extensions developed in Chapter 3). The network added capabilities by enabling the health subsystem to include operator and safe zone risk analysis. These values can either be updated in real-time or can use pre-determined values (e.g. a risk level for the operator given current skill level). Further, multiple hypotheses were demonstrated for the safety zone analysis, enabling the UAS to choose the lower risk zone in a manner consistent

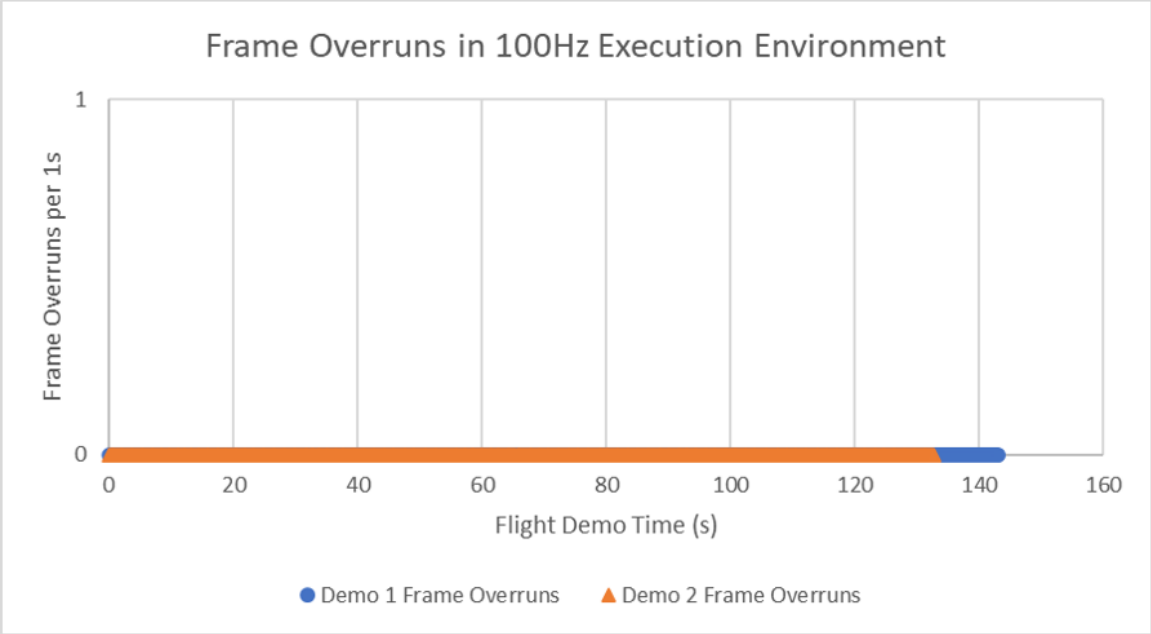

Figure 5.10: Analysis of frame overruns for the two flight demonstrations. Frame overruns are defined as each time the computing cycle takes longer than the time allotted in the 100Hz fixed frame update. As seen in the plot, there were zero frame overruns during both flight demonstrations.

with the rest of the risk analysis. This analysis was underpinned by the network training based on the extension developed in Chapter 3, which allows the network to be adapted as new information becomes available. This mechanism is critical for a UAS ecosystem which provides the necessary information to understand the risk relationships. Finally, the full network can run on a flight computer that powers small UAS, as demonstrated in this chapter, thereby applying to the full UAS ecosystem. This application of this research demonstrates the capability to provide a quantitative risk analysis that supports risk analysis frameworks like SORA [2] currently being pursued by regulatory agencies. Further, since the risk analysis network is easily expandable to greater details, the network has the capability to utilize data from algorithms previously developed, in development, and under future development as detailed in Chapter 2. This capability provides a new baseline, and Section 5.7 discusses the extensions to the full ecosystem.

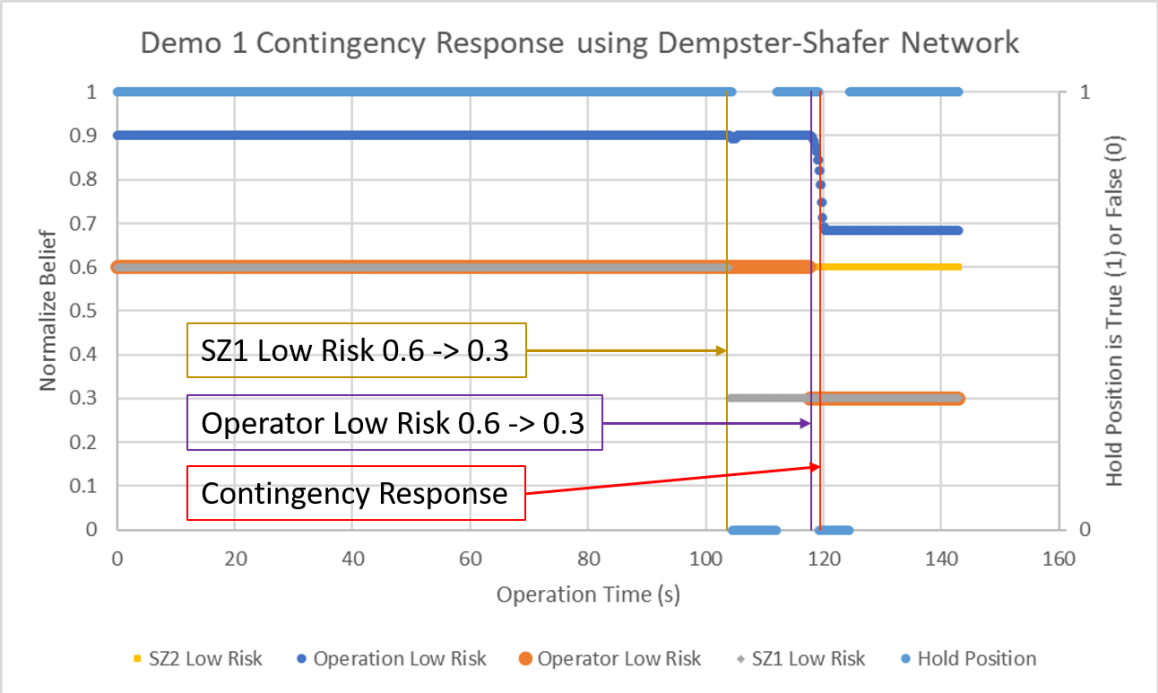

Figure 5.11: Flight demonstration one of Dempster-Shafer network risk evaluation with real-time decision-making onboard a small UAS. The reduction in low risk for the primary safety zone (SZ1), which signifies an increase in medium/high risk for that safety zone, results in the UAS deciding to switch safety zones to the secondary safety zone. During this maneuver, the UAS continues the mission since the resulting operation risk is sufficiently low. The reduction in low risk for the operator, which signifies an increase in medium/high risk for the operator, results in the UAS deciding to land since the operation risk is too high to continue the mission.

5.7 Extensions

5.7.1 Application to UAS Ecosystem

The real-time decision application in this chapter assumes sufficient evidence through operation results to learn the transitions between nodes in the network. Per the discussion in Chapter 2, risk analysis needs to be applied to UAS operations before hundreds or thousands of flight hour results are available. Through the environment discussed in Chapter 2, this data is available. Figure 5.13 provides a more complex risk analysis DS network which can leverage common data across various systems and environments. One important point to note here is how the network scales with respect to increasing complexity. While this

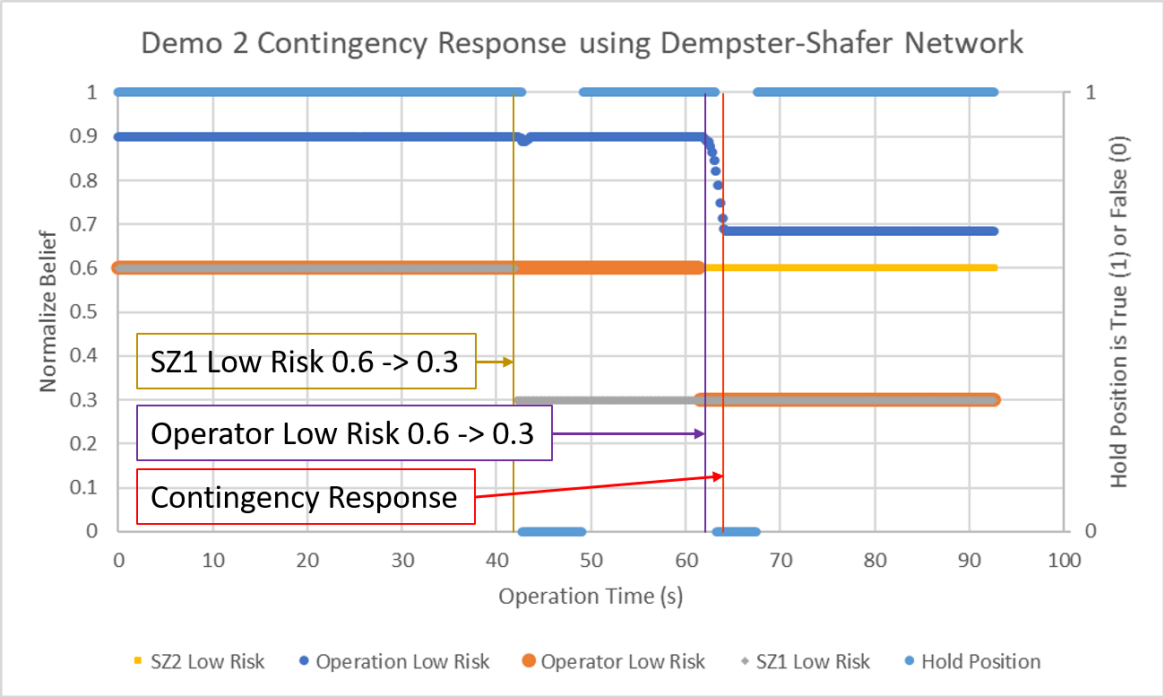

Figure 5.12: Flight demonstration two of Dempster-Shafer network risk evaluation with real-time decision-making onboard a small UAS. The UAS responses in this demonstration are consistent with demonstration one in Figure 5.11, showing that the overall system is repeatable in its responses.

topic is discussed in terms of computation time in Section 3.6.5, a more pressing concern is with respect to interactions between θs. Structure is used to increase explainability and decrease computation time, but at the expense of removing the possibility of interactions between specific θs. For example, including θs (a, b, c, and d) in a node allows unlimited interactions of the probability distribution between those θs. If it is assumed that (c and d) are conditioned on (a and b), then the two sets of θs can be separated into two nodes with a joint conditional belief matrix relating them. However, that removes the possibility that (a and b) can also be conditioned on (c and d) in a directed, acyclic graph. For a higher complexity network, unknown relationships between θs may drive the need to include more θs in the same node, resulting in slower computation times. These consequences result in a design cycle — including more θs initially in the same node and observing the belief marginal distribution in that node over time may lead to breaking that node into multiple nodes with

conditional relationships due to certain interactions between θs being effectively zero in all pertinent situations.

However, data flows and evidence data retention operate differently in a multi-vehicle context with multiple simultaneous operations compared to a single-vehicle context. Previous data retention methods discussed for training in Sections 4.3.2 and 5.3 use Murphy's rule [19] and the weighting from Section 3.5 to capture episodic data, maintaining information across extended periods. This method assumes, across similar episodes, all previous data is captured in a single, consistent network, allowing nodes such as the Operation Risk node to retain information. For example, the combined data retained in the Operation Risk node is still equal to the combined data in all other nodes, such as the Navigation Risk node, multiplied by the transition matrices and combined as evidence into the Operation Risk node. The environment discussed in Chapter 2 breaks this model. Figure 5.14 shows one potential model of the UAS environment. Compiled data can be stored in many datastores within this environment. For example, insurance agencies would likely compile data within their risk models of flight and/or pilot/operator behaviors. Regardless of where the data is stored, data is compiled at different rates as operation hours are built up for UAS, operators, and various environments.

In order to evaluate the risk for a particular operation, the appropriate data must be pulled from each of the datastores, entered into the risk network, such as in Figure 5.13, and calculated to determine the overall risk of the operation. Decisions are then made regarding risk transference and maximum allowable risk. Once the operation is complete, the data is used to improve the relationships in the network. Herein lies the difference from the single system model previously used. The operation risk is no longer a history of data captured in Murphy's Rule [19]. Instead, it is a combined set of data calculated through the network based on multiple datastores with difference amounts of retained evidence, making Mur-

phy's Rule [19] unusable. In order to get past this issue, the Evidential Reasoning Rule [89] is used, which, like Dempster's Rule [1], does not require the evidence history to add new evidence, but still enables weighting the new evidence based on the hours of operation associated with the new evidence. The evidence combination result will not be retained directly, but will be used to update the appropriate episodes to improve the relationships learned in the Figure 5.13 network. As such, the results will be retained through the transition relationships and the datastores for retained data such as the various environment datasets, UAS datasets, etc.

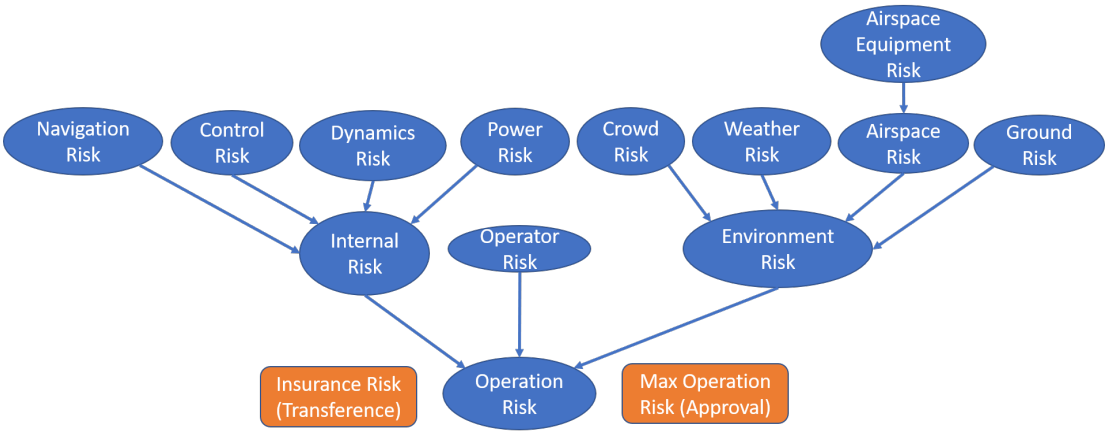

Figure 5.13: Dempster-Shafer network for the UAS ecosystem risk analysis. This network is similar to the network in Figure 5.2, but it is more complex to include various features which could be considered common across operations. For example, the risk of hitting the ground (Ground Risk) in a given area of operations is likely to be common across operations in that area and could leverage previous research to estimate the effects of ground impact [14]. Likewise, the same DJI [7] platform models could leverage common data in the Internal Risk node while common autopilot navigation systems could leverage common Navigation Risk information. Decisions for operations are shown in orange as the acceptable risk transference (a question of insurance) and the acceptable risk level (a question for the regulatory agency).

5.7.2 Distributed Analysis

As a result of these interactions and independent datastores, platforms and stakeholders will be performing distributed analyses, which may result in different outcomes. A contemporary example is getting quotes from multiple insurance agencies for driver's insurance. Dif-

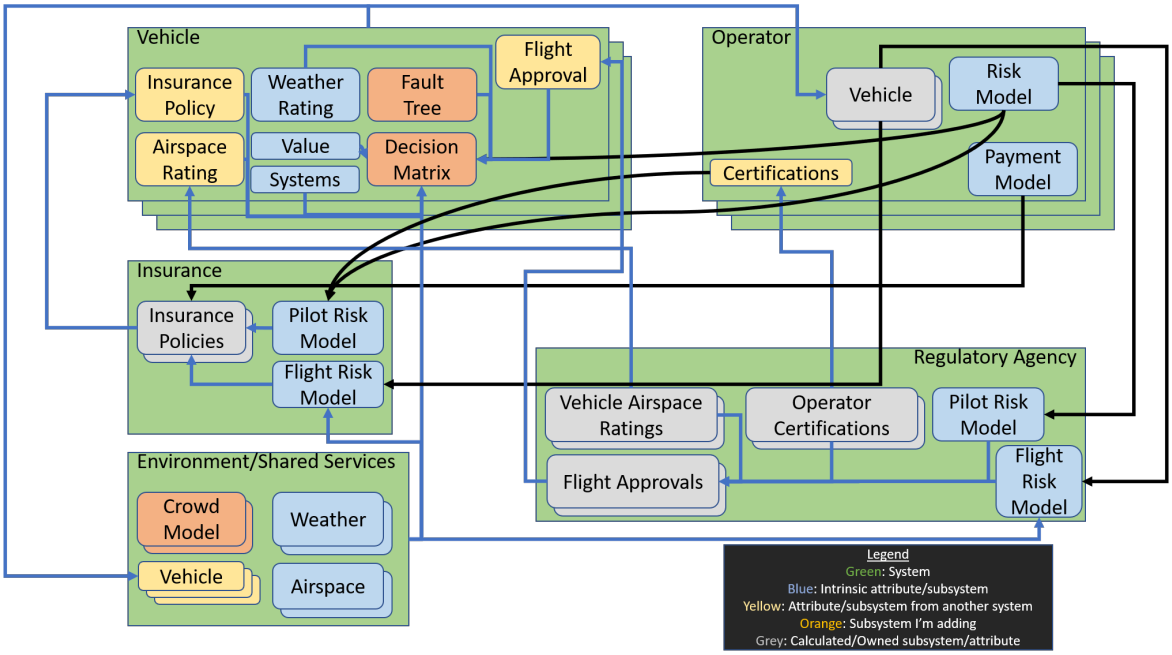

Figure 5.14: A model of the UAS environment, including many of the factors that would impact risk evaluation of operations and the relationships among the various systems in the environment. This model incorporates five major systems: the UAS which perform operations, operators which execute operations with the UAS, insurance agencies which insure the UAS operations, a regulatory agency which ensures safe UAS operations, and the environment in which the UAS operate. Each of these systems interact in multiple ways, and the data flows depicted in the model enable the risk analysis, which each of the systems — other than the environment model — perform. The insurance agencies use models of the operator and flight risk and reward to determine operation premiums. The regulatory agency uses models of the operator and flight risk to determine whether the risk is within maximum acceptable risks. Operators use risk and reward models to determine whether they are willing to pay the insurance premiums required to operate. The UAS uses operation risk models to determine real-time risks of various operation profiles to inform the operator or make automatic decisions.

fering information and risk assessments may result in different premium quotes. Persons seeking the lowest premium for the same risk profile tends to balance these quotes, driving the premiums to similar values for similar risk profiles. In the same way, interactions in the UAS ecosystem will tend to drive distributed analyses towards the same results for similar information, whether through direct communication or "indirect communication" (i.e. dynamic effects within the UAS ecosystem). Analyses can also be purposefully distributed to reduce computation or required information sharing. For example, as marginals perco-

late as evidence through the risk-analysis network, those marginals encapsulate the entirety of the evidence upstream in the directed, acyclic graph. In Figure 5.13, the marginals for the "Internal Risk" node can be seen to encompass the risk assessments for the navigation risk, control risk, power risk, and dynamics risk. Consequently, purposefully distributed computations can be performed with the UAS risk analysis passing the internal risk to the agency assessing the risk, which combines that data with another system's assessment of the environment risk. Thus, the design of the network not only facilitates information sharing between operations, but also facilitates how the computations are distributed based on data access and storage capabilities.

CHAPTER 6

CONCLUSION

Regulatory agencies are moving towards risk-based approval frameworks for UAS access to air space [11]. Governments have mandated that these agencies find a way to safely integrate UAS into the air space [26]. How that integration is done is still a work in progress as the necessary infrastructure, such as remote ID [11], is being developed, per the UAS Safety Symposium notes in Appendix F. The safety symposium held in August 2018 as part of this research provided a strong basis of understanding for this environment. While risk assessment is currently qualitative, this model is unsustainable as the proliferation of UAS means there will be too many approval requests to evaluate qualitatively. Thus, a quantitative mechanism is required which supports and integrates with a risk analysis framework. Much effort has been put into research for these quantitative mechanisms (Section 2.6), yet most suffer from either a limited operational focus or a mechanism that requires too much *a-priori* data, such as with the Bayesian Belief Networks. Dempster-Shafer Theory [1] [3] is commonly applied to risk analysis yet suffers from high computational requirements, often exceeding what is available on small UAS. Valuation networks [16], of which Dempster-Shafter networks are a subset, offer an option which could be computationally feasible for small UAS but require *a-priori* information for the joint mass distributions.

This research pulls together those various threads by developing extensions to Dempster-Shafer networks that enable training the network from evidence inputs. These evidence inputs take the form of operational results that are fed back into the network to improve the understanding of the relationships between nodes. Chapter 3 develops these extensions and experimentally shows that the training algorithm is capable of learning the joint mass

distributions (the transition potentials in the Shafer and Shenoy terminology [17]) to a level that provides more understanding of the system than joint mass distributions as typically entered by experts. These extensions were applied to an autonomous car scenario in Chapter 4 and demonstrated to perform better overall than baseline deceleration profiles and a Bayesian Belief Network approaches. Finally, these extensions were applied to the UAS risk analysis problem in Chapter 5. Simulation testing demonstrated that this risk analysis method outperformed the baseline system and enabled additional capabilities for the UAS to respond to changing external conditions in real-time. Further, flight demonstration showed that this system is capable of running on small UAS in real-time. Extensions were then discussed to apply this research to the full UAS ecosystem. In summary, a quantitative risk analysis framework capable of capturing long-term data for understanding risk relationships and applying that to real-time decision making on small to large UAS has been developed and demonstrated through this work. This quantitative risk analysis framework could serve as the numerical mechanization of the risk-based approval frameworks being developed for regulatory agencies to integrate UAS into air space.

6.1 Recommendations

The following research trajectories are suggested to extend this work:

- Test the UAS risk decision criteria at various points on the ROC [106] curve to evaluate the trade-offs between false alarms and missed detections.
- Use the UAS ecosystem to train the Dempster-Shafer network, as described in Section 5.7.
- Test the UAS ecosystem with real-world data.
- Increase the complexity of the UAS ecosystem to understand the interactions between multiple regulatory and insurance agencies

- Test various UAS risk analysis networks including the one proposed in Figure 5.13 to understand the implications of granularity of the network and how information can be combined at various levels.
- Improve the episodic learning by adding statistical triggers for the episodes, as mentioned in Section 3.4.3. Statistical tests should be added to determine when episodes are sufficiently different to warrant new episodes, and the same tests should be used to allocate updated data to existing episodes when episodes already exist for the current data. This addition enables the training to progress with no direct human input; only human oversight would likely be necessary to ensure appropriate training progression.
- Implement and compare other optimization techniques, as described in Appendix C, for the DS network training.
- Integrate current research and algorithms cited in Section 2.6 into this risk analysis model. This step includes taking work such as the ground collision models that provide impact estimates and integrating these risk models into the quantitative risk model developed in this research to have a fully modeled and data-driven risk analysis.

The addition of these research trajectories would result in a system capable of updating on its own with human supervision, but no human interaction required during the updates. Furthermore, the resulting system would use cutting edge, data-driven models to feed the evidence inputs for this risk analysis network, enabling accurate analysis of the risks, which combine to form the full operation risk analysis. Missing or unproven models have significantly higher unknown masses, resulting in high possibilities (plausibilities in Dempster-Shafer terms) for each of the high, medium, and low operational risk estimates, thus making operation approval significantly less likely. The beauty of this system is that it gracefully handles refinement of these analysis models while using the same approval

criteria. Finally, the UAS ecosystem would enable the risk analysis model to much more quickly incorporate data from similar or identical UAS platforms, operation scenarios, etc., resulting in significantly faster network training.

Appendices

APPENDIX A

DEMPSTER-SHAFER MULTI-PARENT REVERSE SOLVER RESTRICTION VALIDATION

In this appendix, we develop the result necessary to enable restricting the inputs to Dempster's Rule and ECR that enable calculating a valid solution to the reverse solver for multiple parents. The following assumption still applies: for a valid solution to be guaranteed, there must be at least as many evidence inputs as parent marginals for which the algorithm is solving, not including evidence inputs that contain all mass in the universal set.

Furthermore, the following simplifying assumptions enable easier analysis while still applying to the general result:

- ECR is only valid for this result when all weights and reliabilities are equal, thus reducing ECR to Dempster's Rule.
- The result is only required to be developed for two evidence inputs. This assumption is because both Dempster's Rule and ECR add the new evidence into the combined result of the old evidence. Thus, if the combined result provides a valid reverse solution method, then combining any new evidence using the same restrictions will in itself be a combination of two evidences under the same restrictions, resulting in a recursive proof.
- This result is only shown for a three option evidence set (a, b, c) and the associated power set. The reason for this can be seen through the solution method developed in Section 3.3.3. Each solved mass only depends on the masses of higher ambiguity which apply to the solution mass. Thus, the shown proof is, again, recursive.

Recall from Section 3.3.3, the child marginal a in terms of two identical sets of parent marginals is given by Equation A.1.

$$a_c = a_p^2 + 2 * a_p \left[(a,b)_p + (a,c)_p + (a,b,c)_p \right] + 2 * (a,b)_p (a,c)_p \tag{A.1}$$

To solve for a_p, the quadratic formula is used to give Equation A.2.

$$\begin{aligned} a_p = - \left[(a,b)_p + (a,c)_p + (a,b,c)_p \right] + \\ \sqrt{\left[(a,b)_p + (a,c)_p + (a,b,c)_p \right]^2 - \left(2 * (a,b)_p (a,c)_p - a_c \right)} \end{aligned} \tag{A.2}$$

From Equation A.2, it is quite clear that a valid, positive solution will be obtained provided that Equation A.3 holds.

$$a_c \geq 2 * (a,b)_p (a,c)_p \tag{A.3}$$

However, this solution specifically applies to the cases in which the two parent evidence sets are not identical. Thus, from Table 3.4, a_c can be calculated through Equation A.4 where $()_1$ denotes the first evidence set and $()_2$ denotes the second evidence set.

$$\begin{aligned} a_c = a_1 * a_2 + ((a,b) + (a,c) + (a,b,c))_1 * a_2 + \\ ((a,b) + (a,c) + (a,b,c))_2 * a_1 + (a,b)_1 * (a,c)_2 + (a,b)_2 * (a,c)_1 \end{aligned} \tag{A.4}$$

Likewise for the left side of the inequality, the following relationships hold:

$$(a,b,c)_p = \sqrt{(a,b,c)_c} \tag{A.5}$$

$$(a,b)_p = -(a,b,c)_p + \sqrt{(a,b,c)_p^2 + (a,b)_c} \tag{A.6}$$

Recalling further from Table 3.4 the following two relationships:

$$(a,b)_c = (a,b,c)_1 (a,b)_2 + (a,b,c)_2 (a,b)_1 + (a,b)_1 (a,b)_2 \tag{A.7}$$

$$(a,b,c)_c = (a,b,c)_1 (a,b,c)_2 \tag{A.8}$$

With substitution, Equations A.4 - A.8 result in Equation A.9.

$$\begin{aligned} &2\Bigg(-\sqrt{(a,b,c)_1 (a,b,c)_2} \\ &+\sqrt{(a,b,c)_1 (a,b,c)_2 + (a,b,c)_1 (a,b)_2 + (a,b,c)_2 (a,b)_1 + (a,b)_1 (a,b)_2}\Bigg) \\ &\Bigg(-\sqrt{(a,b,c)_1 (a,b,c)_2} \\ &+\sqrt{(a,b,c)_1 (a,b,c)_2 + (a,b,c)_1 (a,c)_2 + (a,b,c)_2 (a,c)_1 + (a,c)_1 (a,c)_2}\Bigg) \leq \\ &a_1 * a_2 + ((a,b) + (a,c) + (a,b,c))_1 * a_2 + \\ &((a,b) + (a,c) + (a,b,c))_2 * a_1 + (a,b)_1 * (a,c)_2 + (a,b)_2 * (a,c)_1 \end{aligned} \tag{A.9}$$

Next, we simplify the left side of Equation A.9 slightly. Note that $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$. Therefore, Equation A.10 holds, and Equation A.11 can be used to constrain the left side

of Equation A.9.

$$\begin{aligned}\sqrt{(a,b,c)_1(a,b,c)_2+(a,b,c)_1(a,b)_2+(a,b,c)_2(a,b)_1+(a,b)_1(a,b)_2} \leq \\ \sqrt{(a,b,c)_1(a,b,c)_2}+ \\ \sqrt{(a,b,c)_1(a,b)_2+(a,b,c)_2(a,b)_1+(a,b)_1(a,b)_2}\end{aligned} \tag{A.10}$$

$$\begin{aligned}2\Bigg(-\sqrt{(a,b,c)_1(a,b,c)_2} \\ +\sqrt{(a,b,c)_1(a,b,c)_2+(a,b,c)_1(a,b)_2+(a,b,c)_2(a,b)_1+(a,b)_1(a,b)_2}\Bigg) \\ \Bigg(-\sqrt{(a,b,c)_1(a,b,c)_2} \\ +\sqrt{(a,b,c)_1(a,b,c)_2+(a,b,c)_1(a,c)_2+(a,b,c)_2(a,c)_1+(a,c)_1(a,c)_2}\Bigg) \leq \\ 2\left(\sqrt{(a,b,c)_1(a,b)_2+(a,b,c)_2(a,b)_1+(a,b)_1(a,b)_2}\right) \\ \left(\sqrt{(a,b,c)_1(a,c)_2+(a,b,c)_2(a,c)_1+(a,c)_1(a,c)_2}\right)\end{aligned} \tag{A.11}$$

Thus, if we show that the right side of Equation A.11 is less than or equal to the left side of Equation A.9, then Equation A.9 holds, and the restrictions will satisfy the constraint necessary to ensure a solution to the reverse solver. To do this, we introduce the restrictions from Equations 3.9 and 3.10. Further, without loss of generality, we assume that $(a,b)_1 \geq (a,b)_2$, $(a,c)_1 \geq (a,c)_2$, $(a,b)_1 \geq (a,c)_1$, and $(a,b)_2 \geq (a,c)_2$.

Using Equation 3.9 for a_1 and a_2 the least constraining choices of $(a,b)_1$, $(a,b)_2$, $(a,c)_1$, and $(a,c)_2$, the right side of Equation A.9 can be constrained via Equation A.12, which also includes a square and square root to end up with the same form as the right side of

Equation A.11.

$$\begin{aligned}
&((3\,(a,b)_1\,(a,b)_2 + (a,b,c)_1\,(a,b)_2 + (a,b,c)_2\,(a,b)_1 \\
&\quad +2\,(a,c)_1\,(a,b)_2 + 2\,(a,c)_2\,(a,b)_1)^2)^{\frac{1}{2}} \leq \\
&\quad a_1 * a_2 + ((a,b) + (a,c) + (a,b,c))_1 * a_2 + \\
&((a,b) + (a,c) + (a,b,c))_2 * a_1 + (a,b)_1 * (a,c)_2 + (a,b)_2 * (a,c)_1
\end{aligned} \tag{A.12}$$

Using the restrictions from Equation 3.10, we cancel terms and end up with Equation A.13.

$$\begin{aligned}
&2\left(\sqrt{(a,b,c)_1\,(a,b)_2 + (a,b,c)_2\,(a,b)_1 + (a,b)_1\,(a,b)_2}\right) \\
&\left(\sqrt{(a,b,c)_1\,(a,c)_2 + (a,b,c)_2\,(a,c)_1 + (a,c)_1\,(a,c)_2}\right) \leq \\
&((3\,(a,b)_1\,(a,b)_2 + (a,b,c)_1\,(a,b)_2 + (a,b,c)_2\,(a,b)_1 \\
&\quad +2\,(a,c)_1\,(a,b)_2 + 2\,(a,c)_2\,(a,b)_1)^2)^{\frac{1}{2}}
\end{aligned} \tag{A.13}$$

Therefore, the original inequality in Equation A.3 is true, and the restricted set is shown to have a valid reverse solution via the method developed in Section 3.3.3.

APPENDIX B

DEMPSTER-SHAFER ALGORITHM WEIGHTING MODIFICATIONS

Other than the ECR method [89], Dempster's Rule and rules created to replace/improve Dempster's Rule consider all evidence equally, without weight. Due to the modifications in Section 3.8, it was necessary to introduce weights to algorithms which could be used in the Dempster-Shafer network. Murphy's Rule [19] and the two Rayleigh methods [20] were easily modified to include weight by weighing the evidential inputs. For Murphy's Rule, this equates to a weighted average. For the Rayleigh methods, this means that evidential masses are multiplied by the associated weights when combined with the other evidential masses, whether they are supportive or in conflict. Zhang's Rule [18] had to be handled differently since multiplying the input evidence masses by the weights do not effect the result because the first use of the evidence masses is to calculate conflict. The conflict calculation uses a dot product, which is normalized, thus removing the weighting. Instead, the weighted average credibility of the original reliability was modified to use both the weighting developed through Zhang's support calculations and the input evidence weights, thus affecting the final result through the evidence input weights.

APPENDIX C

DEMPSTER-SHAFER TRANSITION SOLUTION ALTERNATE METHODS

The solution to solve the transitions in a Dempster-Shafer network, as detailed in Section 3.3.2, was chosen to be least squares with modifications to the solution when the non-negative constraint on all transition values was violated. Alternate solution methods are available that could satisfy the design requirements of low computation requirements and fast solutions. In particular, two options were discussed and are recommended for future evaluation.

- Moore-Penrose Pseudo-Inverse [110]. The definition of the transition in the Dempster-Shafer hypertree, as given in [17], provides the child-to-parent transition values using the transpose of the parent-to-child transition values. This is potentially not the only available definition. An inverse is impossible, given that many cases result in a non-invertible matrix. Figure 3.5 is a simple example of this case. However, the generalized inverse, using the Moore-Penrose definition, is not restricted in the same manner. This method was not chosen for current development in order to use the hypertree definition given in [17]. It is recommended in future analysis to compare the Moore-Penrose pseudo-inverse definition with the transpose definition in the Dempster-Shafer hypertree and evaluate which method performs better and provides more intuitive and explainable results.

- Iteration using the separation principle [102]. The separation principle is a well-known concept in control theory that allows the system observer to be designed separately from the system controller because the observer dynamics are sufficiently faster than the controller dynamics that the two parts of the system are effectively

decoupled. Similarly, the Dempster-Shafer network is typically updated at a slow rate relative to the update rate of a real-time system. For example, the network is often updated on the order of 1Hz to 10Hz, while real-time controllers usually update significantly faster. Given the relatively slow update rate, it should be possible to design a dynamic transition solution method which has significantly faster dynamics than the network update. Again, it is recommended in future analysis to compare this solution method with the least squares solution method developed in this paper.

APPENDIX D

TRAFFIC LIGHT SCENARIO INITIAL CONDITIONS

The initial conditions for the traffic light scenario are listed in this appendix for repeatability of tests.

Table D.1: Traffic light scenario initial conditions. This table lists the initial conditions for the traffic light scenario.

Initial Condition	Value	Initial Condition	Value
Driver Speed Mean (m/s)	15.0	DS Observation Rate (s)	0.2
Driver Speed StDev (m/s)	4.0	DS Observation Window	5
Cross Traffic Speed Mean (m/s)	16.0	Cross Traffic Expected Speed Mean (m/s)	16.0
Cross Traffic Speed StDev (m/s)	6.0	Cross Traffic Expected Speed StDev (m/s)	2.0
Cross Traffic Density Mean (veh/m)	20.0	Cross Traffic Expected Density Mean (veh/m)	30.0
Cross Traffic Density StDev (veh/m)	8.0	Cross Traffic Expected Density StDev (veh/m)	5.0
Cross Traffic % Speed up at Yellow	5	Cross Traffic % Continue at Yellow	20
Cross Traffic % Slow down at Yellow	75		
Visibility Range (m)	0.0	Visibility	1.0
Visibility Range (m)	20.0	Visibility	0.6
Visibility Range (m)	200.0	Visibility	0.5
Yellow after Sim Start Mean (s)	9.0	Yellow after Sim Start StDev (s)	6.0
Red after Yellow Mean (s)	6.0	Red After Yellow StDev (s)	1.0
Green after Yellow Mean (s)	2.0	Green after Yellow StDev (s)	0.5
Cross-Walk Activated Probability	0.8		
Cross-Walk Blinking before Yellow Mean (s)	5.0	Cross-Walk Blinking before Yellow StDev (s)	0.5
Cross-Walk Visibility Mean (m)	20.0	Cross-Walk Visibility StDev (m)	5.0

APPENDIX E

DEMPSTER-SHAFER NETWORK TRAINING RESULTS

Detailed results of training are provided in this appendix to show comparisons between various methods. Multiple methods were evaluated with the general trend that the more precise application of training episodes to transitions resulted in the transitions capturing the nuances of relationships between the Dempster-Shafer combined evidence distributions in the parent and child nodes. In controls terminology, this phenomenon can be characterized through observability. All transition potentials are being calculated through modified least squares optimization based on the combined distribution at the parent and child nodes. A single distribution for each of the parent and child nodes is effectively a single basis vector with the transition potentials being the function that spans the space between those vectors, as show in Equation 3.1.

The move to episodic learning in Section 3.4 results in the parent and child node combined distributions being represented by multiple statistically different distributions or basis vectors. The addition of the weights in learning the potentials in Section 3.4.1 forces the transition potential matrix to learn based on each basis function, capturing the improved observability. In Section 5.3 this method is further refined by only applying the episodes to nodes and transitions that are significantly affected by the episodes, the two steps of which are shown in Figures E.4 and E.6. As described in Section 3.4.3, the logical extension to this method is to calculate statistical significant differences in combined evidence distributions at each node to determine which nodes and transitions are affected by each episode. Differences in captured information through the transitions potentials are shown in Figures E.1 through E.6. Note that both of these methods have room for improvement. Some in-

formation can be captured early, as shown in Figure E.2, while later changes should be minimized, as shown in Figure E.6.

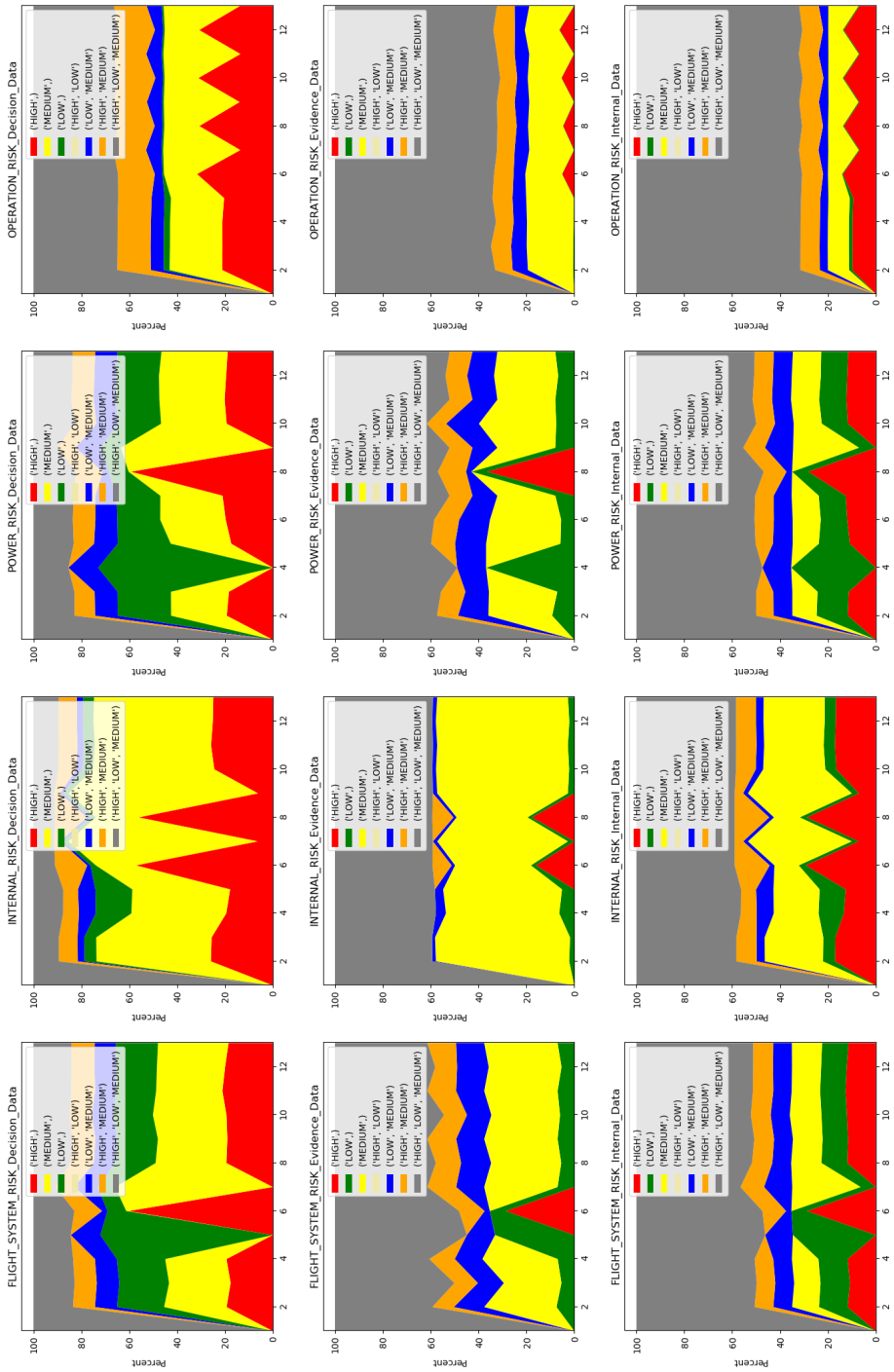

Figure E.1: Training episodes for the UAS scenario in Chapter 5 applying all episodes to all nodes. The x-axis represents each episode update. This figure focuses on combined evidence distributions for the nodes affected by vehicle failure and failsafe rates. In each of the three rows of plots, multiple updates can be seen with the same distributions suggesting that those data points aren't adding new information or basis functions to the transition potential learning algorithm.

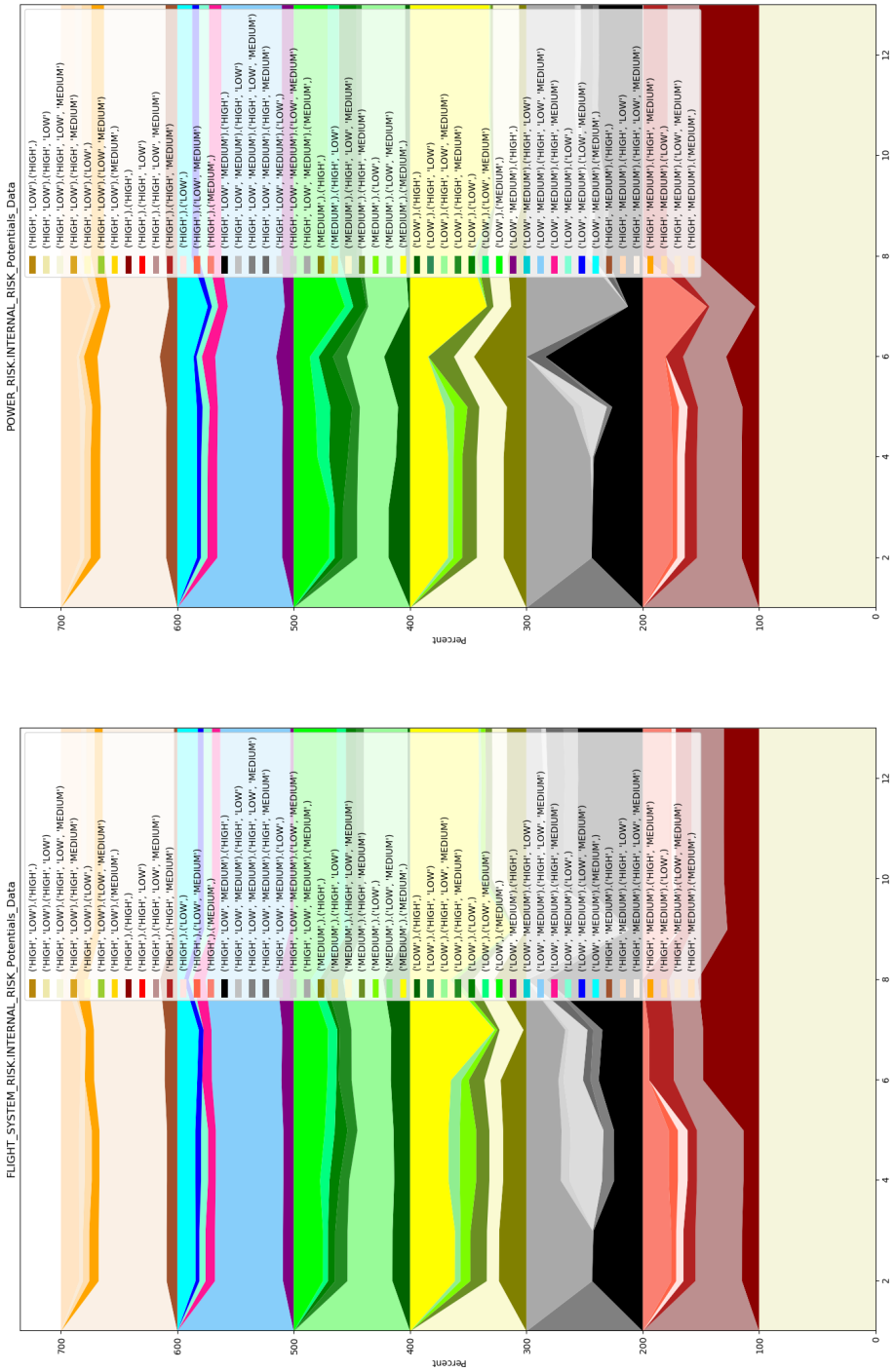

Figure E.2: Training episodes for the UAS scenario in Chapter 5 applying all episodes to all nodes. The x-axis represents each episode update. This figure focuses on the transition potentials updates between the two vehicle systems nodes and the overall vehicle systems node (the Internal Risk node). Of interest in comparing this figure with Figure E.1 is that initial updates to all the potentials are occurring as early as the first update, even though Figure E.1 shows that information available in update one only applies to a small subset of the transition potentials (a single basis function).

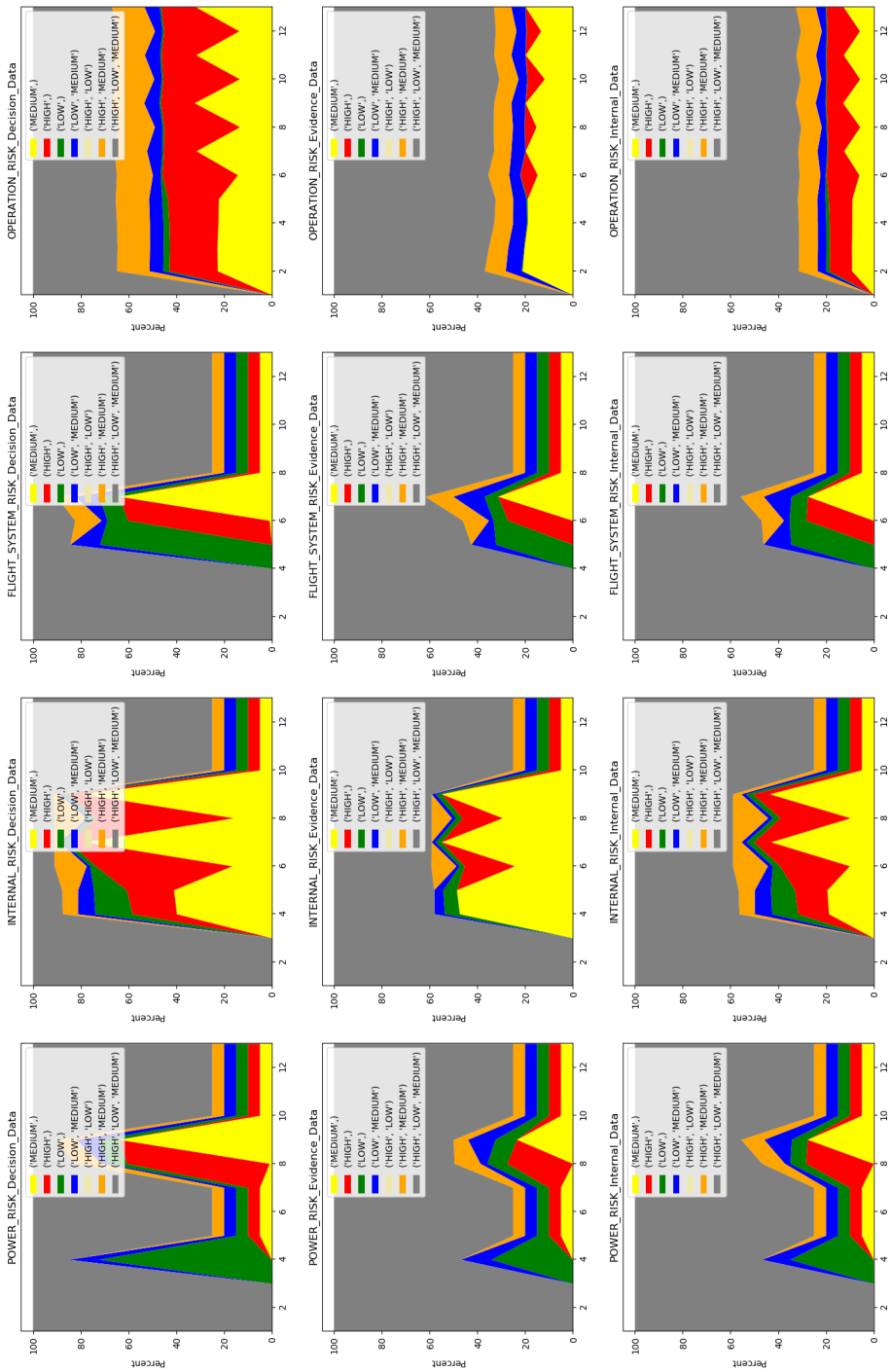

Figure E.3: Training episodes for the UAS scenario in Chapter 5 applying each episode to distributions that are affected by that episode. The x-axis represents each episode update. This figure focuses on combined evidence distributions for the nodes affected by vehicle failure and failsafe rates. For each distribution, changes in the distribution can be clearly seen in the episodes which directly affect that distribution, while subsequent episodes retain enough information to minimize loss of data in the transition potentials.

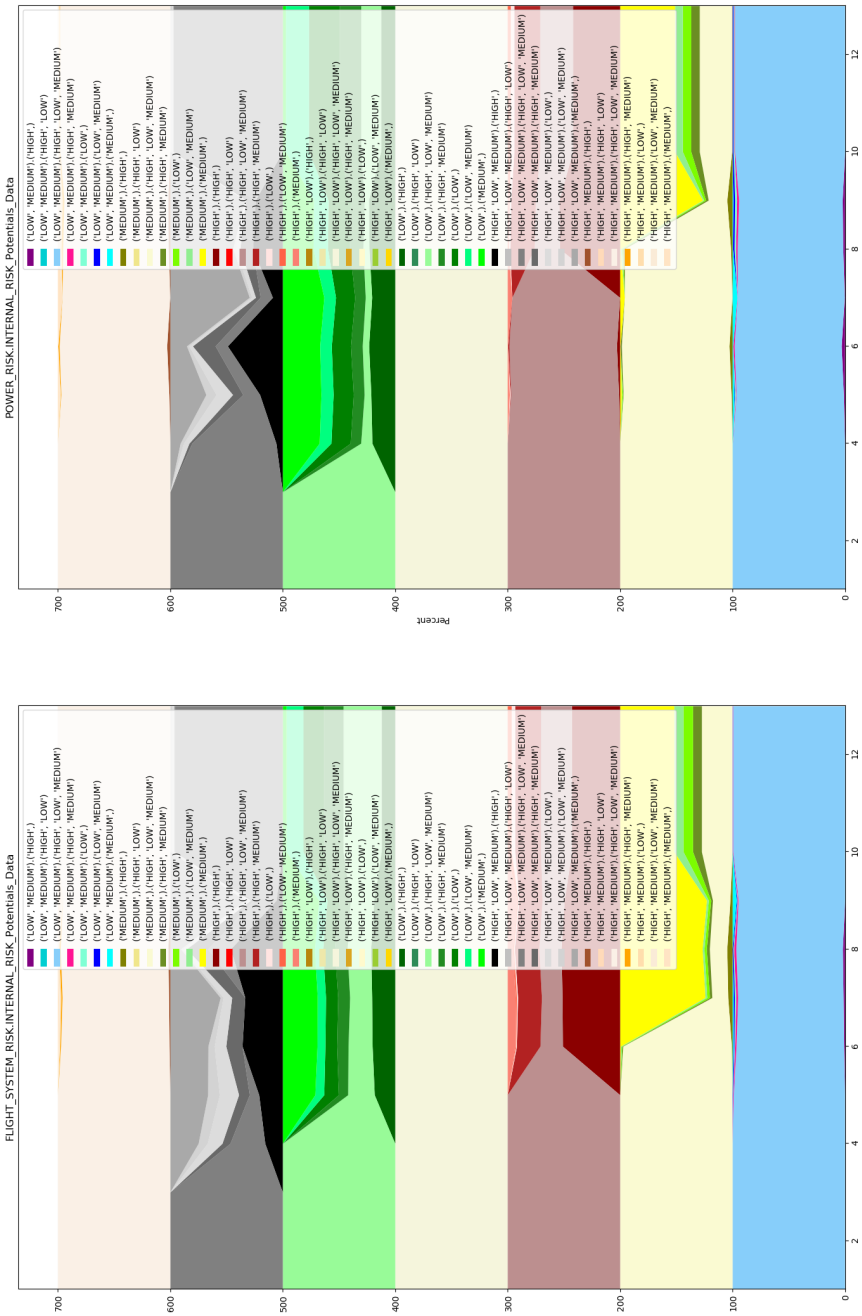

Figure E.4: Training episodes for the UAS scenario in Chapter 5 applying each episode to distributions that are affected by that episode. The x-axis represents each episode update. This figure focuses on the transition potentials updates between the two vehicle systems nodes and the overall vehicle systems node (the Internal Risk node). Comparing this figure to Figure E.2 shows a significant difference in learning behaviors. The transition potentials in this figure only start updating once information is available that directly affects these potentials, and changes to the transition potentials after episodes directly affecting these potentials are minimized.

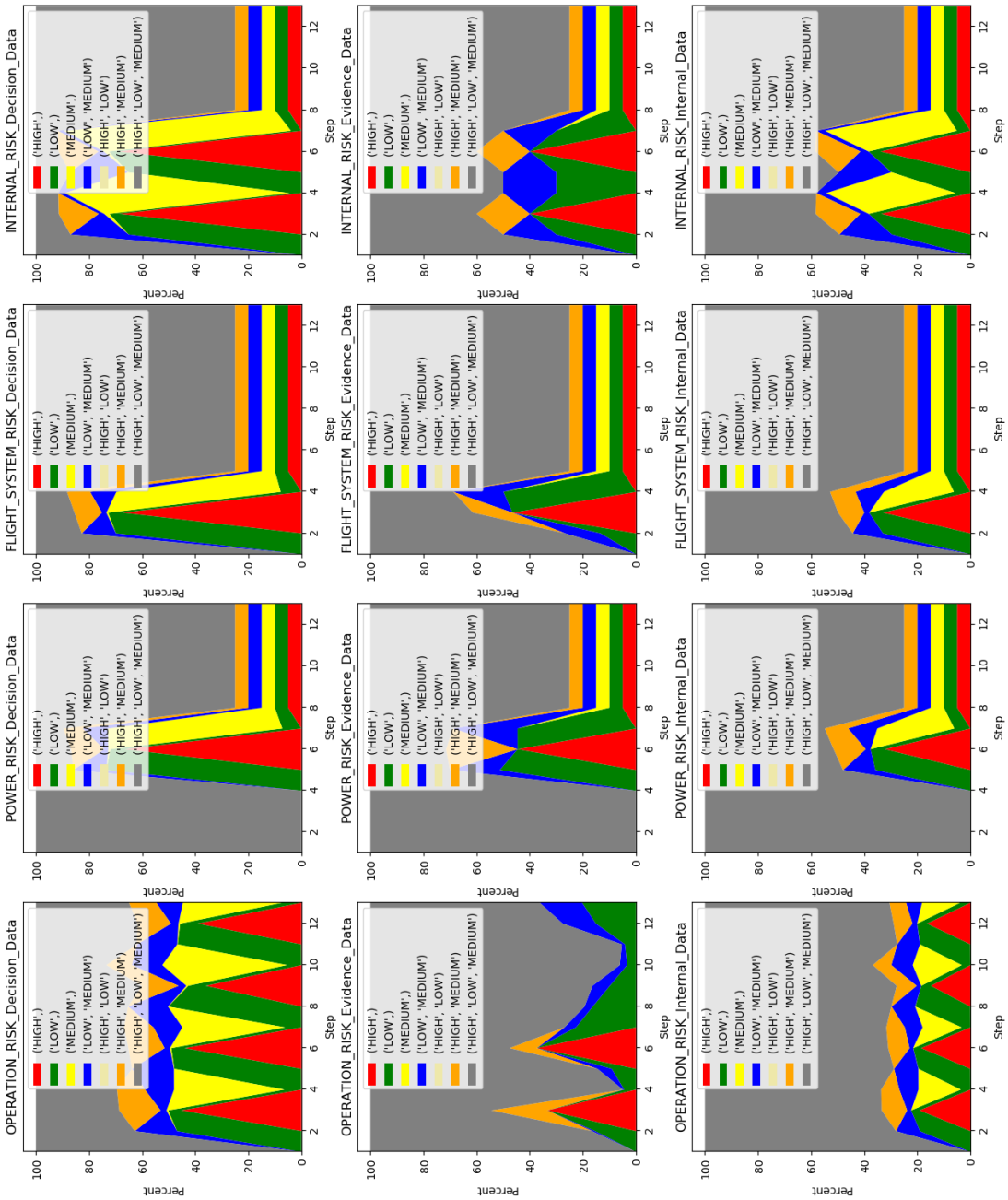

Figure E.5: Training episodes for the UAS scenario in Chapter 5 applying each episode to distributions that are affected by that episode and only training the transitions of multi-parent nodes that are affected by each episode. The x-axis represents each episode update. This figure focuses on combined evidence distributions for the nodes affected by vehicle failure and failsafe rates. For each distribution, changes in the distribution can be clearly seen in the episodes which directly affect that distribution, while subsequent episodes retain enough information to minimize loss of data in the transition potentials.

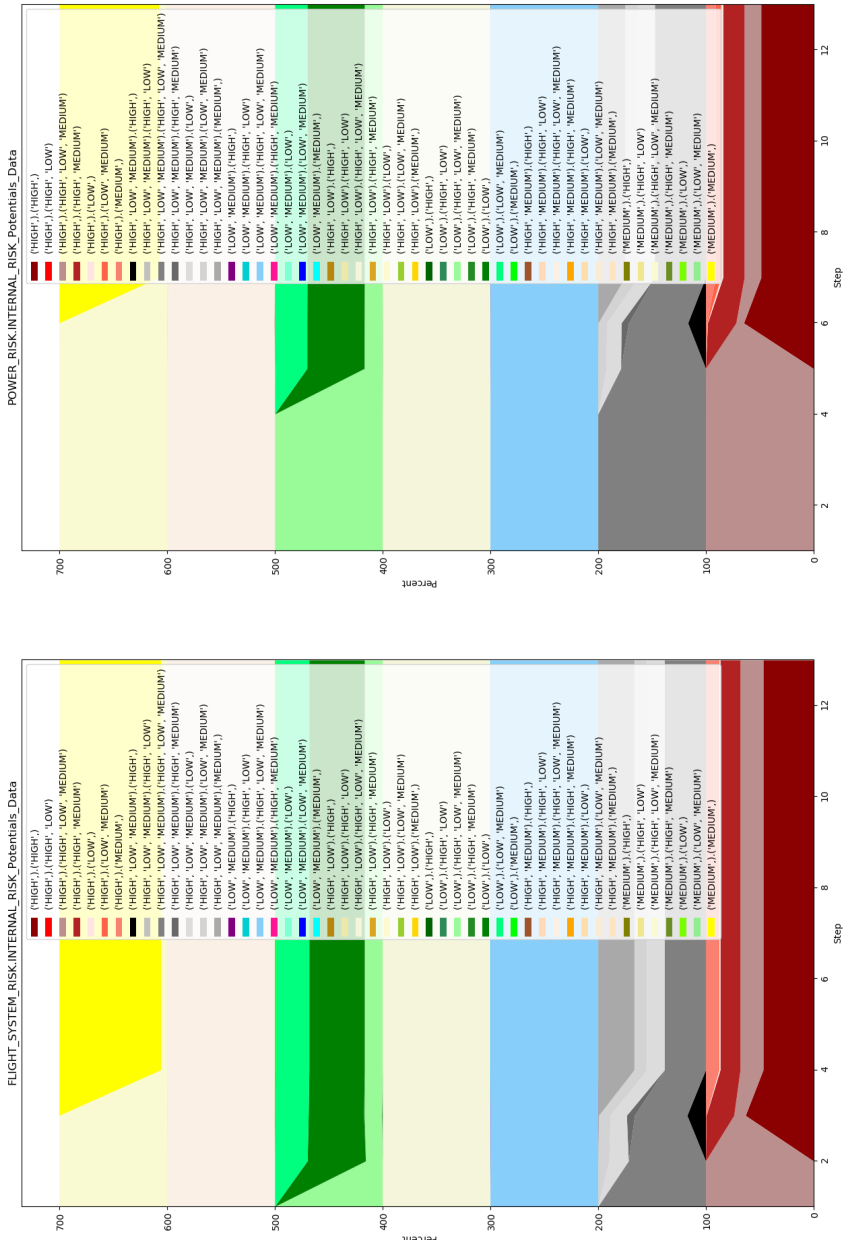

Figure E.6: Training episodes for the UAS scenario in Chapter 5 applying each episode to distributions that are affected by that episode and only training the transitions of multi-parent nodes that are affected by each episode. The x-axis represents each episode update. This figure focuses on the transition potentials updates between the two vehicle systems nodes and the overall vehicle systems node (the Internal Risk node). Comparing this figure to Figure E.4 shows a noticeable difference in learning behaviors. The transition potentials in this figure update in the same pattern between the Flight Systems Risk to Internal Risk transition and the Power Risk to Internal Risk transition. Figure E.4 shows a different behavior between the two, even though the training inputs in this scenario were identical.

APPENDIX F

SAFETY SYMPOSIUM RAW NOTES

Date: 7/23/2018

F.1 Summary/Major Takeaways

- There is a decided lack of information in this space, but there are several areas and methods of information outreach available by a variety of different enterprises.
  - www.faa.gov/uas
  - www.faa.gov/go/waiver
  - https://faadronezone.faa.gov portal
  - 14 CFR Part 1.1 and 107
  - 49 USC 40102
  - FAA Advisory Circular 107-2 Small Unmanned Aircraft Systems
  - Social media platforms (Facebook, Instagram, YouTube, Twitter, and others)
  - AMA
  - UAST (Unmanned Aircraft Safety Team)
  - Drone Advisory Committee
  - Commercial Drone Alliance
  - AUVSI

  - Waiver Safety Educational Guidelines
  - Public Law 112-95 and subsequent reauthorization bills for FAA
  - Pathfinder, Public Safety Partnership, Integrated Pilot Project and other industry-FAA coordinated agreements
  - FAASafetyTeam: FAASafety.gov
  - uashelpdesk
  - FAA AFS-800 Policy Branch
  - Variety of conferences, shows, symposiums, conventions and meetings
- Much of the industry, especially on the regulations and law side, are in a wait-and-see mode for the precedents to be set, usually when forced by a major incident
  - Out of a million registered users, there are 100,000 commercial operators with a sUAS Remote Pilot Certificate registered to operate for compensation or hire, as well as Public Aircraft Operations (government function)
- The biggest issue with enforcing regulations and prosecuting offenders is identification of the UAS and operator: agree, but working on a rule that will provide policy.
- Operator education is essential and is currently severely lacking: agree

F.2 Notes from Presentation 1 — The Rise of Drones and Insurance

- Reinsurance handles 65% of recoverables from non-US companies – a method of spreading risk
- Farm mutuals rent drones to operators for surveying land, soil, blight, etc. – typically

fixed wing craft are used.

- Real Estate is another application
- Legal issues could be faced / how insure UAS operations for those?
- Everything is new with drone – drones, operators, and the law
- Law is always behind tech, but is catching up: My comment at all my presentations is that lawyers and the insurance companies will drive the industry more than the FAA, unless there is blood spilled, in which case Congress will get involved.
- FAA – finalized the current drone rules as of 8/29/16
- True, and have added Remote Pilot certificate testing and recurrent testing, as well as an Advisory Circular, Inspector Guidance, Test Sites, academic coordination with Center of Excellence (ASSURE), and a variety of pilot programs for evaluating operations of drones and risk analysis
- Drones are a federal affair, because the feds are the primary regulators of national airspace
- There have been drone incidents, often involving manned aircraft
- The rules from the FAA lowered the barriers to entry for operators
  - Previous rules: required a pilot's license: a certificate is required for all pilots of all aircraft
  - Current rules: 14 CFR Part 107 requires a "license" and applies to operators flying for compensation or hire. (FAA still does require one for drones: the Remote Pilot Certificate, for those who want to operate for compensation or hire under Part 107. Recreational pilots who do not subscribe to a community based set of guidelines fall under jurisdiction and enforcement of 107, although – no pilot's license ("certificate") is necessary if the drone meets the requirements

of:

  * Must fly for hobby/recreational use only
  * Operate under a community based set of safety guidelines
  * $\geq$.55lbs and $\leq$ 55lbs
  * Class G airspace only ($\leq$ 400' AGL typically)
  * Maintains Visual Line of sight
  * Avoids manned aircraft
  * Does not endanger the National Airspace
  * Notifies airports and ATC when within 5 miles
  * Per NDAA 2018, must register as operator, using registration number for all owned aircraft over .55lbs and $\leq$55lbs as recreational user or for each aircraft. If a Remote Pilot operating under 107, then single aircraft per registration number.
  * Request permission to operate in controlled airspace
  * Don't fly impaired
  * Also regulates recreational use of drones per Part 1.1 and 101.

- Under Part 107 as a model aircraft operator:
  * Maximum airspeed of 100mph
  * Cannot operate over people or moving vehicles or emergency response
  * Minimum weather visibility of 3 miles
  * No carriage of hazardous materials
  * Preflight inspection required

    - Daylight hours only unless waived, similar to other conditions requiring waiver

- Most common drone use is renting the drone and operator
- Waivers can be obtained to Part 107: under 14 CFR 107.200 and .205
- Other countries have drone regulations also. Canada and the US are fairly closely synced, but some other countries are more progressive in their regulations
- Drone insurance is a legal issue – Insurability brings in a host of issues:
  - Violation of FAA Rules
  - Physical Damage and Bodily Injury
  - Nuisance laws
  - Trespass laws (One of the primary reasons for drone violations)
  - Invasion of Privacy (The other major reason for drone violations)
    - Private individuals
    - Government use/searches
  - Stalking and harassment
  - Wiretap laws
- Boggs vs Meredith – 2015 – where does private property end and public airspace begin?
  - With drones, FAA can regulate down to the blades of grass, if exposed to outside.
  - If indoors, it's not airspace, and the FAA doesn't regulate it
- ISO drone endorsements

- CGL (Commercial General Liability = A standard insurance policy issued to business organizations to protect them against liability claims for bodily injury (BI) and property damage (PD) arising out of premises, operations, products, and completed operations; and advertising and personal injury (PI) liability.) policy – look to if there's a loss
- Drones as a service – drone highways
- Insurers have reasonable pricing models, but these are dependent on information from the operators.
- Some insurers use exclusions to keep operators within the bounds they specify
- Global Aerospace removed exclusions because crashes invariably break at least one exclusion regardless of the operation.
- The current model is supposed to be on a per flight basis (based on the forms). Often, it is not executed this way
- Insurance tends to be evaluated in one of two ways:
  - It falls into a "normal" bucket. In this case, it is often passed along to the re-insurers who insure/price it based on standard rates. Typically, this is fully automated/computerized with human oversight.
  - Some parameters of the insurance request are outside the norm. In this case, it is typically evaluated by a human in the primary insurance companies who helps to determine the risk model and pricing

F.3 Notes from Discussion One

F.3.1 Group 1

(Bullet points are statements made during a dialogue)

- What is there for regulation and insurance of re-certification?
- Probably if within the scope of work, covered under underwriting questions with input of engineer, probably coverage issued based on underwriting risk audit
- No standards of safety versus known vulnerabilities, but a risk management profile could be developed
- Too new – law hasn't caught up with tech. Drone may need certain characteristics. Geo fencing = no access for drones that are equipped with this feature
- DJI implements altitude restrictions, but can be bypassed
- Authorized first demo at airport – DJI "bricked" because the system shut down. It recognized that it was in restricted space, and there was no way to bypass that restriction. DJI has since updated its software.
- Risk mitigation – there are workarounds
- Business friend in Korea – when too close to the DMZ, the internal GPS will not start
- How can you run self-diagnostics before traveling to an authorized place to fly if the system won't start up outside of the authorized location?
- the definition of "safe" catches up to tech
- safety is piecemeal, developed based on experience and reaction/over reaction
- The UCF team has analyzed data on how people are using UAVs. FAA said not going to fund that work.

- Court case – FAA does not have the right to regulate hobbyists
- Parker – Pre 107, since superseded
- Hobbyists were "refunded" for UAS registration fees, although the money wasn't returned, and 107 was fixed
- FAA on 107 commercial side have authority to regulate, no ability to regulate "hobbyists" guidelines: this isn't true re: hobbyists (model aircraft operators). We regulate through Part 1.1 and Part 101 directly, and through Part 107 if modeler doesn't comply with a community-based set of guidelines
- There are flight restrictions that apply to UAS as well
- Helicopters and UAS operate in similar airspace, which creates potentials for issues
- Regulations are for safety. Pathfinder program – fly drones out of LOS and at night
- The airport had a 2 day safety risk assessment with airlines. Identify risks with operations
- Beyond LOS analysis: higher concern is UAS and helicopters because the occupy the same airspace. UCF analysis estimated the well-clear distance, which went into the FAA guidance. They got a 333 exemption for their flying.
- How can you stay "well-clear" when you don't know what's out there? How do you prove you know what's out there? Definition of "well-clear" is key.
- Assume worst case – they have a few years of helicopter data from Boston, built simulations. There are no flight plans for helicopters.
- Put transponders on UAS?
- It's going that way
- Look at new technology on cars

- Look at the UTM Systems
- Application information on FAA website. Regulation data there also.
- Blocked finding info about the impact of an impact. What happens in a drone strike? Likelihood of damage, shutdown, how shut down. Wind turbulence of wing hit another plane?
- 1st UAV strike this past year was into a helicopter
- NTSB investigated. The drone pilot had no idea the helicopter was there because drone was BVLOS at the time and operator walked away
- White house lawn drone. Operator had no idea where the drone went down – lost contact with it.
- Regulations cannot rely on operators having full control: not sure what this means, but disagree. We want operators to have control. That is a main focus. Note: Joel Dunham discussed this with the FAA representative after his comments. The point was made with respect to degraded situations in which the UAS may have to take over some level of control and respond appropriately, given that many UAS now have a large disconnect between the operator and the low-level controls that fly the UAS. The FAA representative's response was that the FAA would likely mandate a level of training and understanding by the operator such that even in a degraded situation the operator would know how the UAS would likely respond (such as standardized return-to-home in the future) and be able to predict and plan for the UAS responses. In that way, the operator still has a level of control over what the UAS will do for the operation.
- Strike tests/simulations have been done. There is a fair amount of data. Ask Hartfield-Jackson International Airport representative: how do they enforce the law?
- Depends on who within the Atlanta Police Department answers. They rely more

on peeping tom laws. No local ordinances/restrictions. "Master's" got restrictions. Most rely on nuisance laws. Video: consent needed from both people. Example: drone outside a high rise videoing – drone collected as evidence due to video

- Trespass suit? Violate federal law?
- Law cannot chase technology. It chases underlying issues. Cobb county wanted to protect Suntrust Park and sent a letter to the FAA. FAA said "stay in your lane. The air is not your lane".
- Devices interfere with drones
- Video and video collection – when does this violate laws? When is this useful for demonstrating regulation compliance?
- Depends on use. There is an expectation of privacy
- On UCF campus, cannot lift off drone from campus, but can take off from across the street and fly over campus
- The airport is the same. May get waiver, but not through airport screening. How do they enforce hobbyists? How do you control trespass in a restricted area?

F.3.2 Group 2: Subject: Safety

- There are two types of "Safety". Actual safety [which seemed to be understood as safety from physical harm] and perceived safety. Perceived safety has been shown to vary depending on the velocity/size/distance of the drone. The primary determinant is velocity.
- The research into perceived safety has been done by interdisciplinary teams and has been determined using measures of changes in skin conductivity, head tilt and heart rate. [this seems to equate perceptions of safety to a physical fear/flight response]

- The measures are specific to the individual as each person has differing physiological responses based on their life experiences.
- Other measures discussed for perceived safety included possible violations of presumed privacy and fear of unknown factors having to do with the drone operations (e.g. who is the operator and what is their intent)
- Perceived safety violations are more of a driver of reports to law enforcement and complaints to regulators than actual safety violations.
- Actual safety can be impacted by being designed into the product, perceived safety can also be designed into the product, the team discussed above used assistance from a product design and art team to make drones look more "friendly".
- Actual safety would include improvements in system redundancy. This can be limited in application due to the payload and weight requirements.

F.3.3 Subject: Safety Regulatory Structure

- Current structure based on weight seems to be the most logical way to divide classes of drones. This makes the model similar to other regulatory classes (autos, aircraft).
- Perhaps additional divisions of the regulatory structure into more weight classes of drones could be useful as the categories are pretty broad right now with each weight class required to be manufactured with additional redundancy and safety characteristics to mitigate the risk.
- Additional segmentation for operators could influence regulation and insurance based on skill level and experience of the pilot, also composition and design of the drone (e.g. a frangible airframe)
- Question for further discussion, should drone operators be "type rated" as pilots are for large commercial aircraft? Do different drones and the control software have

vastly different flight and operational characteristics?

- Belief of the panel: Commercial use will drive regulations. Regrettably, losses will come first and drive public opinion which will influence/prioritize new regulations. The industry is a wait and see mode until a loss experience occurs. Agree.
- Other comments:
  - There are a lot of different redundancy issues for the vehicles and software that need to be addressed in the safety arena.
  - It may require regulation at the manufacturer level to regulate some safety issues. [product liability?]
  - Is it possible to regulate drone operations through the software?

F.3.4 Some Assumptions

- An underlying assumption for panel discussions was that this applied to UAS operators who want to follow the law/do the right thing.
- The balance on regulating through software is between education/testing/restrictions and convenience. Because we assumed that operators want to do the right thing, regulating through software could be effective. However, as with many of the checklists in software, too much results in lack of convenience and good actors sidestepping the rules.
- This is perhaps where operator certification could apply, as a certification number validated through the system could bypass training and tests.
- One of the big reliability issues with software is the cost – as manufacturing processes become more rigorous, cost increases quickly, building barriers to entry.
- Law expert's inputs and discussions included the assumption of malicious actors (as

he has experience with South Carolina Corrections and the use of drone by prisoners). In this case, how to deal prevent and deal with incidents is significantly different.

F.3.5 Pathway for Operations beyond Part 107

Currently waivers applications are online. Responses to waiver requests usually take about 90 days. Once a critical mass of waivers is reached in a specific area (e.g. night flights) the FAA may change the rules or there may be some standardization of responses. Also airspace authorizations are key, with LAANC coming on line will help a lot.

F.4 Notes from Discussion 2

F.4.1 Cases

1 A UAS videoing a major foot race crashes, causing the death of some runners

2 UAS down a passenger jet by destroying engines, causing loss of the jet and loss of life

3 A UAS flying without authorization prevents a fire fighting tanker from entering the fire zone, causing millions of dollars of property damage in a California urban area

F.4.2 Group 1 discussion on Scenario 2

(Bullet points are statements made during a dialogue)

- There is a loss of communication. Should have been geofenced.
- Everyone gets sued – the airline, manufacturer, UAS operator
- Statutory liability for passengers. Everyone gets sued. Point of proximate cause. Tender defense to insurance companies – probably settle.

- Airport contractors require additional drone coverage.
- All drone operations should have liability coverage
- Heightened training requirements if near an airport, additional hours, contractual requirements. Experience requirements for hobbyists. There are waiver applications. This goes back to risk management – was the UAS pilot properly experienced?
- How would they know the experience?
- Ask at training (the software asks for the certification level of the operator, number of hours, etc)
- DJI UAS log flight hours
- City keep log of drone close calls
- FAA has sightings database – most entries are bogus. Sightings are reported to police and perhaps to the FAA. Study of UAV pilots focused on weather. If not a pilot, don't focus on weather.
- Small UAS are too cheap to worry about crashing, so they don't care.
- Behaviors better, but not trained
- Congress said cannot regulate hobbyists and probably most drones out there.
- As UAS get cheaper, more are buying them
- More manufacturers also (like GoPro)
- GoPro failed due to power loss issues – they had to work out the kinks on the first aircraft
- More manufacturers mean cheaper UAS
- What about the use of collision avoidance systems

- Must be bigger and heavier
- Put transponder in UAS for ADSB to alert planes/helicopters? Add collision avoidance? Technology will advance and likely become smaller
- Sensor array to ID the pilot must be within the infrastructurer
- Question to Hur about avionics devices running
- The LiPo batter was a big development for avionics
- The money going into technology is enormous. Leads to breakthroughs in smaller and more powerful batteries.
- Fuel cells demonstrated with fixed wing. Proof of concept shown years ago.
- Syria: small UAS being used to drop grenades
- Most legal issues with companies will settle due to reputational issues
- Peachtree road race scenario – who carries insurance coverage that is enough?
- Anyone involved in the race would be sued – City, PRR, UAS
- Aerial Photography – Additional coverage for UAS?
- Human risk managers review to ensure appropriate coverage for applications
- Often churches, etc are running races. Are they permitting UAS? Checking coverage?
- Probably not covered
- Large loss with power transformers?
- Frequency and severity issues – people don't follow the laws

F.4.3 Moving Cases through the Court System

- The hardest part of the process right now would be (for case #2) finding enough parts of the drone equipment to identify the operator.
- For cases #1 and #3, many of the unsafe operations are being traced by postings to social media.
- A third issue is finding the "proof" and being sure that the evidence follows proper chain of custody procedure. E.g. destroying the SIM card where flight data is stored, worries that the logged track could have been changed if not in safe custody.
- Often a drone "violation" (trespassing, unsafe operations) are called in to local law enforcement as first responders. As drone operations are federally regulated, this is usually a dead end. Due to law enforcement response times, the drone and the operator are long gone by the time they arrive.
- Other issues with pursuing a case into the court system, in addition to finding the drone, tracing the operator, and getting the correct authorities involved, once those parties are in place, if there are damages, often the recovery of damages must be allocated between the operator, the manufacturer and the victim. The victim may need to pursue recovery from their own insurance company for damages similar to being hit by an uninsured motorist.
- On the state level, depending on the state law, the percentages of responsibility will be allocated by a jury. (Negligence theory). Some states have a system that if the majority of the fault does not lie with the perpetrator then the victim does not recover.
- Can this be changed with regulations? Another issue raised was enforcement of the regulations.
- Question: Is insurance a driver or reactor to the drone operations environment? Yes, both.

- There are still many gaps in the regulations that will get decided on a case by case basis.
- Disconnection of direct physical control of the drone as an object impacts perceived safety.

F.4.4 Pathway beyond 107

(Bullet points are statements made during a dialogue)

- Folks working on beyond visual LOS. 1st person goggles (FPV = First person view) or spotter. They mapped UCF campus using 4-5 spotters. That's no longer allowed. Could ADSB be put in? Not possible for many systems.
- Hobby associations will push back on an ADSB requirement
- Hobby stay in G airspace. Limit them to 400ft.
- Risks of wind gusts, GPS failures, must harden systems against inappropriate misuse. Too easy to leave/stray out of approved airspace
- US done over Iranian airspace was taken over electronically.
- Hacker controls it.
- Hijack collision avoid systems. Spoof system by transmitting fake vehicle reports

F.5 Presentation 2 by Dr. Vela – UAS Statistics

- Most flight data is acquired from flight tracking websites to which the operators post their flights.
- The websites are usually shut down soon after discovering that their data is being scraped.

- Amazon Web Services (AWS) generally has pretty unsecured data.
- The data presented covers probably 25-30% of all drone flights – which shows thousands of flights per day.
- Americans are, in general, pretty good about following regulations. Usually around 1%-1.5% of drone flights (max) fall outside the regulations. These calculations are based on out of visual range, too close to airports, too high of altitude, etc.
- Note that more of the flights are likely out of visual range due to obstructions in view (trees and such), but this can't be calculated effectively in this data set (yet).
- Altitude tests (how high can I go) are usually the first test attempted by new UAS operators, and usually go over 400ft in open areas, which is illegal.
- 5% of flights are 500+ feet and within 5nm of an airport

F.6 Presentation 3 - Malicious use of UAS

- There are not currently any good ways to stop malicious actors from using UAS. The biggest problem is still identification.
- A drone can be flown to a prison carrying many cell phones in a small container, grabbed by a prisoner from a window, and be gone before the guards arrive. The cell phones will already be distributed, and if the guards find the operator, it is typically a kid who was paid to fly it and has no idea where the money came from.
- There are desires to block cell phone and UAS control signals around prisons.
- Blocking UAS signals has been discussed around airports/non-prison settings as well. However, if the blocked signal forces a lost-comms scenario, there is no guarantees of what the UAS will do. Will it land (is it in a safe place to land)? Will it return to base?

- Taking control of a UAS by regulating authorities through a signal has also been discussed. This is similar to the flight baggage model in which all locks have a master lock that TSA can use to open and check. In the case of UAS, this has the potential issue of malicious actors gaining access to these "backdoors".

F.7 Discussion 3

F.7.1 Goals of the Parties

- Desired advancements? Improved software, awareness of other air traffic for avoidance, issues with "line of sight" operations due to the environment.
- Big/difficult current issue is: What are the rules?
- Operators (recreational, light or entry-level commercial) may not know where to find out. Some education by manufacturers is in place, unknown how much should be their responsibility.
- Controllers (FAA towers, airspace controllers) may not communicate with operators, may not know their responsibility.
- Law enforcement (local police, sheriff, etc) may not know rules.
- Much more education in this space is needed.
- Updates to flight management software in this area might help.
- Huge need for uniformity of access and review of accident reports, incident reports.
- Need for uniformity and standardization in NOTAMS and TFRs to speed dissemination and comprehension by non-professionals.
- Question, should the minimum requirements for operator certification be increased? Will that lead to less compliance?

- Desired advancement in software and hardware: Lost signal guidance (what happens if...)
- Additional safety enhancement / damage mitigation planning for commercial operations.
- Ideas
  - "safe zones" for drones to pilot to in case of problems
  - trouble broadcast frequencies and codes similar to aviation's "guard" frequency and emergency codes used in both aviation and marine
- Idea: Regulators maintaining a listing of authorized or certificated pilots, perhaps for only commercial operators. We already do.
- Much advancement is waiting for commercial operators to take up. Many improvements are not yet required but may come as there is consolidation. Much is waiting for industry maturity.
- Issue: General Aviation and helicopters are not tracked for common flight tracking when flying VFR if not requested. Flight following is available, but there is tracking if filed IFR.

F.7.2 Operators

- Other traffic data desired – currently can get ADSB from a website, but not immediately
- Must be careful where point their camera
- Educational issue for operators training – they don't know what is illegal in many cases: Very true.
- Do operators understand privacy? Cognitive develop process and how it applies here?

Do they just not know? Can educational videos help?

- Fun education training exercises for users – when power up a system the first time.
- But purchasers may not be the users
- Similar to the issues for lasers and planes
- Notifications need to be provide to users about warnings – real time versus delayed and GPS accuracy are issues
- What do commercial users need?
  - Single source for rules/laws: FAA provides regulations; lawmakers provide statutes/law.
  - Where are no fly zones, etc: Available on FAA App
  - Can they takeoff/land in national parks? No, due to Department of Interior jurisdiction
- Getting airspace authorization from towers – only have to alert, not get authorization now
- Language is not standardized and is not user friendly. Aviation language is standardized for air traffic and pilots, but new additions are part of the new world of UAS.
- Commercial users and/versus hobby users – what do they need to know?
- Database of operator history and experience – there are companies that are set up/looking into developing databases, but they are struggling in this. Due to privacy issues, but will become more transparent with FOIA
- The industry needs to mature.

F.8 Discussion Notes 4

F.8.1 Insurance and Lawyers

- Desired future of UAS.
- Primary: How to explain the issues, operations, etc to a jury.
- Methods for identification of the parties (operators, equipment).
- Protection from the destruction of data.
- Pilots providing data for research, possible economic incentives?
- Underwriting and claims are the two main touch points when insurance companies will get the best data to price their products.
- Unintended consequences of providing and receiving data and loss history, same for regulation. The need for a feedback loop to correct the unintended consequences.
- Impact of the hacking and mods culture getting around limitations. Encryption limiting the ability.
- Current regulations allow pilots to self-report to NASA/NTSB for accidents and incidents. Are drone pilots aware of this? What information is important to gather?
- Membership organizations: AOPA, AMA, UAS coalition, current major influencers of policy, lobbyists, RC clubs all influencing the space and might be avenues for dissemination of education.
- There might be resistance as some answers the organization's members don't want to receive.
- There is a high level of trust in the manned aviation community for accident reports leading to improvements in aviation instead of punishment. Could this work in the drone community?

- What problem are we trying to solve for: Human deaths seem to be the measure and driver of change, but more due to shock of "mass" casualties (aviation accidents) versus one-by-one casualties (auto).
- Mental separation between the action and results of the action the farther away from the event in space.
- CNN, research on frangible drones.
- Are innovations better driven from the commercial or manufacturer side?
- Will additional regulations (on manufacturers) drive up the cost of production and kill the market?
- Regulatory certainty allows more progress in the field.
- Uber model: Disregard the law and allow it to catch up to the "new reality". Change management by building a critical mass of change.
- Does safety "sell"? Like in the auto market? Did it always, or is this a recent development? It is always assumed safety is the responsibility of the FAA, so it doesn't sell. No air carrier advertises that they are safer than brand X, because of "minimum requirements" being met to be certified.
- Are the incentives different in the drone market?
- What uses are drones being put to and does this matter for regulations? Yes, due to the construct of the rules, i.e. agriculture, spraying, dropping of stores, carriage of external goods and property of another across state lines, urban taxis, etc.
- Data safety issues? Is someone stealing/hacking the data produced? Can that cause operator harm?
- Can the software be hacked in a major way to cause harm? (e.g. Were airline reservations systems hacked last year when each of the major airlines had "system crashes"

that stranded large numbers of passengers all over the world?) What are the weak points (AWS)?

F.8.2 Insurance and Law

- Drone legal services – AOPA
- UAS Coalition
- Discussion is analogous to firearms, boats, and motorcycles/dirtbikes

F.9 Discussion Notes 5 – Where from Here?

- Issues for research: Remote ID (craft and operator), Key as it is the starting point for much of the enforcement environment.
- New licensing models: Equipment size and function.
- Education of operators on existing regulation now that drones are defined as aircraft. Drones have been defined as aircraft since 2005, not just since Part 107.
- Current education providers with classes for Part 107 licenses.
- Integration with STEM in current school curricula. Design and deployment of new courses.
- Atlanta airport developing systems for use and improvement of waivers for 107 ops at airport.

REFERENCES

[1] A. Dempster, "Upper and lower probability induced by a multivalued mapping," *The Annals of Probability*, vol. 38, 1967.

[2] JARUS, "Jarus guidelines on specific operations risk assessment (sora)," Report, Jun. 2017.

[3] G. Shafer, *A Mathematical Theory of Evidence*. Princeton University Press, 1976, pp. 1–314.

[4] H. Xu and P. Smets, "Reasoning in evidential networks with conditional belief functions," *International Journal of Approximate Reasoning*, vol. 14, pp. 155–185, 2-3 1996.

[5] M. Satell, *Philly by air*, `https://www.phillybyair.com/blog/drone-stats/`, Web Page, 2020.

[6] *Event38 unmanned systems*, `https://event38.com/`, Web Page, 2011.

[7] *Dji*, `https://www.dji.com/`, Web Page, 2018.

[8] U. S. D. o. T. Federal Aviation Administration, *Safety management system (sms)*, `https://www.faa.gov/about/initiatives/sms/`, Web Page, 2018.

[9] ——, *Compliance program*, `https://www.faa.gov/about/initiatives/cp/`, Web Page, 2018.

[10] M. B. Jamoom, M. Joerger, and B. Pervan, "Unmanned aircraft system sense-and-avoid integrity and continuity risk," *JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS*, vol. 39, no. 3, 2016.

[11] *Remote identification of unmanned aircraft systems*, `https://www.regulations.gov/document?D=FAA-2019-1100-0001`, Web Page, 2020.

[12] L. R. Cork, R. Walker, and S. Dunn, "Fault detection, identification and accommodation techniques for unmanned airborne vehicle," Australian International Aerospace Congress, Report, 2005.

[13] P. F. A. D. Donato, "Toward autonomous aircraft emergency landing planning," PhD Thesis, University of Michigan, 2017.

[14] E. Ancel, F. M. Capristan, J. V. Foster, and R. C. Condotta, "Real-time risk assessment framework for unmanned aircraft system (uas) traffic management (utm)," 17th AIAA Aviation Technology, Integration, and Operations Conference, Jun. 2017.

[15] S. Ramasamy, R. Sabatini, and A. Gardi, "A unified approach to separation assurance and collision avoidance for uas operations and traffic management," Miami, FL, USA: IEEE International Conference on Unmanned Aircraft Systems (ICUAS), Jun. 2017.

[16] P. P. Shenoy, "Valuation-based systems: A framework for managing uncertainty in expert systems," *Fuzzy Logic for the Management of Uncertainty (L. A. Zadeh and J. Kacprzyk, Eds.)*, pp. 83–104, 1992.

[17] G. Shafer and P. Shenoy, "Probability propagation," University of Kansas School of Business, Report, 1989.

[18] Z. Zhang, T. Liu, D. Chen, and W. Zhang, "Novel algorithms for identifying and fusing conflicting data," *Sensors*, vol. 14, 2014.

[19] C. K. Murphy, "Combining belief functions when evidence conflicts," *Decision Support Systems*, vol. 29, 1999.

[20] J. Hurley, C. Johnson, J. Dunham, and J. Simmons, "Nonlinear algorithms for combining conflicting identification information in multisensor fusion," IEEE Aerospace Conference, Mar. 2019.

[21] J. Wang, K. Dixon, H. Li, and J. Ogle, "Normal deceleration behavior of passenger vehicles at stop sign–controller intersections evaluated with in-vehicle global positioning system data," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1937, pp. 120–127, 2005.

[22] eBay, *Bayesian-belief-networks*, `https://github.com/eBay/bayesian-belief-networks`, Web Page, 2020.

[23] J. Dunham and E. Johnson, "Unmanned aerial systems health monitoring architecture," IEEE Aerospace Conference, Mar. 2019.

[24] U. S. D. o. T. Federal Aviation Administration, "Summary of small unmanned aircraft rule (part 107)," Federal Aviation Administration, Report, Jun. 2016.

[25] *Uas remote identification*, `https://www.faa.gov/uas/research_development/remote_id/`, Web Page, 2020.

[26] *Presidential memorandum for the secretary of transportation*, https://www.whitehouse.gov/presidential-actions/presidential-memorandum-secretary-transportation/, Web Page, 2020.

[27] ——, *Federal aviation administration program partnerships*, https://www.faa.gov/uas/programs_partnerships/, Web Page, 2018.

[28] *Aim section 2: Controlled airspace*, https://www.faa.gov/air_traffic/publications/atpubs/aim_html/chap3_section_2.html, Web Page, 2020.

[29] A. S. Aweiss, B. D. Owens, J. L. Rios, J. R. Homola, and C. P. Mohlenbrink, "Unmanned aircraft systems (uas) traffic management (utm) national campaign ii," AIAA SciTech Forum, Jan. 2018.

[30] A. Ferguson and J. McCarthy, "Sharing the skies (safely): Near term perspective on suas integration in the nas," Integrated Communications, Navigation and Surveillance Conference (ICNS), Apr. 2017, 3B2–1.

[31] U. S. D. o. T. Federal Aviation Administration, *Section 333*, https://www.faa.gov/uas/beyond_the_basics/section_333/, Web Page, 2018.

[32] B. A. Garner, *Black's Law Dictionary (10th ed)*. Thompson Reuters, 2014, p. 334.

[33] J. Knight, B. Smith, and Y. Chen, "An essay on unmanned aerial systems insurance and risk assessment," 2014 IEEE/ASME 10th International Conference on Mechatronic, Embedded Systems, and Applications (MESA), Sep. 2014.

[34] C. H. Koh, C. Deng, L. Li, Y. Zhao, S. K. Tan, Y. Chen, B. C. Yeap, and X. Li, "Experimental and simulation weight threshol study for safe drone operations," AIAA SciTech Forum, Jan. 2018.

[35] J. Lazatin, "A method for risk estimation analysis for unmanned aerial system operation over populated areas," 14th AIAA Aviation Technology, Integration, and Operations Conference, Jun. 2014.

[36] J. Li, W. Monroe, and D. Jurafsky, "Understanding neural networks through representation erasure," *arXivpreprint arXiv:1612.08220*, 2016.

[37] J. Kolodner, "An introduction to case-based reasoning," *Artificial Intelligence Review*, vol. 6, pp. 3–34, Mar. 1992.

[38] Volocopter, *Volocopter*, https://www.volocopter.com/en/, Web Page, 2018.

[39] A. Meyer, *Xavion*, xavion.com, Web Page, 2018.

[40] A. International, *Standard practice for methods to safely bound flight behavior of unmanned aircraft systems containing complex functions*, https://www.astm.org/Standards/F3269.htm, Web Page, 2020.

[41] I. LAKSHMINARAYAN, "Model-based fault detection for low-cost uav actuators," Thesis, UNIVERSITY OF MINNESOTA, 2016.

[42] R. E. Weibel and J. R. John Hansman, "An integrated approach to evaluating risk mitigation measures for uav operational concepts in the nas," 6957, vol. 2005, Arlington, VA: AIAA 4th Infotech@Aerospace Conference, Sep. 2005, p. 11.

[43] R. E. Weibel and R. J. Hansman, "Safety considerations for operation of unmanned aerial vehicles in the national airspace system," Massachusetts Institute of Technology, Report ICAT-2005-1, Mar. 2005.

[44] R. E. Weibel, "Safety considerations for operation of different classes of unmanned aerial vehicles in the national airspace system," Thesis, MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 2005.

[45] C. W. Lum and B. Waggoner, "A risk based paradigm and model for unmanned aerial systems in the national airspace," Infotech@Aerospace, Mar. 2011.

[46] B. Rattanagraikanakorn, A. Sharpanskykh, M. Schuurman, D. Gransden, H. A. Blom, and C. De Wagter, "Characterizing uas collision consequences in future utm," 2018 Aviation Technology, Integration, and Operations Conference, Jun. 2018.

[47] J. Breunig, J. Forman, S. Sayed, L. Audenaerd, A. Branch, and M. Hadjimichael, "Modeling risk-based approach for small unmanned aircraft systems," Report, Jul. 2018.

[48] N. A. Neogi, C. C. Quach, and E. Dill, "A risk based assessment of a small uas cargo delivery operation in proximity to urban areas," Proceedings of the 37th Digital Avionics Systems Conference (DASC), Sep. 2018.

[49] J. T. Luxhoj, W. Joyce, and C. Luxhoj, "A conops derived uas safety risk model," *Journal of Risk Research*, 2017.

[50] V. Kumar, F. Wieland, S. Toussaint, and J. T. Luxhøj, "Uas (unmanned aerial system) safety analysis model (usam)," 14th AIAA Aviation Technology, Integration, and Operations Conference, Jun. 2014.

[51] R. B. Ferreira, D. M. Baum, E. C. P. Neto, M. R. Martins, J. R. Almeida Jr., P. S. Cugnasca, and J. B. Camargo Jr., "A risk analysis of unmanned aircraft systems

(uas) integration into non-segregate airspace," International Conference on Unmanned Aircraft Systems (ICUAS), Jun. 2018.

[52] E. Denney, G. Pai, and M. Johnson, "Towards a rigorous basis for specific operations risk assessment of uas," 2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC), Sep. 2018.

[53] R. E. Weibel, M. W. M. Edwards, and C. S. Fernandes, "Establishing a risk-based separation standard for unmanned aircraft self separation," Berlin, Germany: Ninth USA/Europe Air Traffic Management Research & Development Seminar, Jun. 2011.

[54] A. Weinert, S. Campbell, A. Vela, D. Schuldt, and J. Kurucar, "Well-clear recommendation for small unmanned aircraft systems based on unmitigated collision risk," *JOURNAL OF AIR TRANSPORTATION*, vol. 26, no. 3, 2018.

[55] RTCA, *Sc-228, minimum performance standards for unmanned aircraft systems*, `https://www.rtca.org/content/sc-228`, Web Page, 2020.

[56] A. Washington, J. M. Silva, and R. Clothier, "A review of unmanned aircraft system ground risk models," *Progress in Aerospace Sciences*, 2017.

[57] E. M. Atkins, A. D. Khalsa, and M. D. Groden, "Commercial low-altitude uas operations in population centers," 9th AIAA Aviation Technology, Integration, and Operations Conference, Sep. 2009.

[58] R. Melnyk, D. Schrage, V. Volovoi, and H. Jimenez, "A third-party casualty risk model for unmanned aircraft system operations," *Reliability Engineering and System Safety*, vol. 124, pp. 105–116, 2014.

[59] J. L. de Castro Fortes, R. Fraga, and K. Martin, "An approach for safety assessment in uas operations applying stochastic fast-time simulation with parameter variation," Winter Simulation Conference, Dec. 2016, pp. 1860–1871.

[60] A. T. Ford and K. J. McEntee, "Assessment of the risk to ground population due to an unmanned aircraft in-flight failure," 10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, Sep. 2010.

[61] C. W. Lum, K. Gauksheim, T. Kosel, and T. McGeer, "Assessing and estimating risk of operating unmanned aerial systems in populated areas," 11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, including the AIA, Sep. 2011.

[62] K. Ijaz, S. Sohail, and S. Hashish, "A survey of latest approaches for crowd simulation and modeling using hybrid techniques," 17th UKSIM-AMSS International Conference on Modelling and Simulation, 2015, pp. 111–116.

[63] W. Kang and Y. Han, "A simple and realistic pedestrian model for crowd simulation and application," *ArXiv*, Aug. 2017.

[64] P. Deville, C. Linard, S. Martin, M. Gilbert, F. Stevens, A. Gaughan, V. Blondel, and A. Tatem, "Dynamic population mapping using mobile phone data," *Proceedings of the National Academy of Sciences*, vol. PNAS Early Edition, Oct. 2014.

[65] P. F. A. Di Donato and E. M. Atkins, "Evaluating risk to people and property for aircraft emergency landing planning," *JOURNAL OFAEROSPACE INFORMATION SYSTEMS*, vol. 14, no. 5, 2017.

[66] M. S. Kaiser, K. T. Lwin, M. Mahmud, D. Hajializadeh, T. Chaipimonplin, A. Sarhan, and M. A. Hossain, "Advances in crowd analysis for urban applications through urban event detection," *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, vol. 19, pp. 3092–3112, 10 2018.

[67] M. Tzelepi and A. Tefas, "Human crowd detection for drone flight safety using convolutional neural networks," 25th European Signal Processing Conference, Aug. 2017, pp. 743–747.

[68] X. Du, A. Dori, E. Divo, V. Huayamave, and F. Zhu, "Modeling the motion of small unmanned aerial system (suas) due to ground collision," *Journal of Aerospace Engineering*, vol. 232, pp. 1961–1970, 10 2018.

[69] J. Vian and J. Moore, "Trajectory optimization with risk minimization for military aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 12, no. 3, pp. 311–317, May 1989.

[70] E. N. Johnson, *Georgia tech uav research facility*, `http://www.uavrf.gatech.edu/platforms/gust/`, Web Page, 2018.

[71] G. Chowdhary and E. N. Johnson, "Adaptive neural network flight control using both current and recorded data," vol. AIAA 2007-6505, AIAA Guidance, Navigation, and Control Conference, Aug. 2007.

[72] S. K. Kannan and E. N. Johnson, "Adaptive control with a nested saturation reference model," American Institute of Aeronautics and Astronautics Conference, Sep. 2003.

[73] S. Twigg, A. Calise, and E. Johnson, "On-line trajectory optimization including moving threats and targets," vol. AIAA 2004-5139, AIAA Guidance, Navigation, and Control Conference, Aug. 2004.

[74] M. J. Logan, J. Gundlach, and T. L. Vrana, "Design considerations for safer small unmanned aerial systems," AIAA SciTech Forum, Jan. 2018.

[75] T. Yomchinda, J. Horn, and J. Langelaan, "Flight path planning for descent-phase helicopter autorotation," AIAA Guidance, Navigation, and Control Conference, Aug. 2011.

[76] G. Patents, *Autorotation guidance command system, device, and method*, `https://patents.google.com/patent/US10124907B1/en`, Web Page, 2020.

[77] M. Hejase, A. Kurt, T. Aldemir, U. Ozguner, S. B. Guarro, M. K. Yau, and M. D. Knudson, "Quantitative and risk-based framework for unmanned aircraft control system assurance," *JOURNAL OFAEROSPACE INFORMATION SYSTEMS*, vol. 15, no. 2, 2018.

[78] S. X. Fang, S. O'Young, and L. Rolland, "Online risk-based supervisory maneuvering guidance for small unmanned aircraft systems," *JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS*, vol. 41, no. 12, 2018.

[79] Y.-J. Lu and J. He, "Dempster-shafer evidence theory and study of some key problems," *JOURNAL OF ELECTRONIC SCIENCE AND TECHNOLOGY*, vol. 15, no. 1, 2017.

[80] E. Feigenbaum, *Knowledge-based systems in japan*, `http://www.wtec.org/loyola/kb/toc.htm`, Web Page, 2019.

[81] J. L. Kolodner, "An introduction to case-based reasoning," *Artificial Intelligence Review*, vol. 6, pp. 3–34, 1992.

[82] J. MacKay David, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003, p. 540.

[83] C. Cortes and V. N. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[84] J. H. Kim and J. Pearl, "A computational model for combined causal and diagnostic reasoning in inference systems," Proceedings of the Eighth International Joint Conference on Artificial Intelligence, 1983, pp. 190–193.

[85] P. Quan, Z. Shanying, W. Gang, and Z. Hongcai, "Some research on conflict and robustness of evidence theory," vol. 27, 4th International Conference on Information Fusion, Nov. 2001.

[86] J. Rogers and M. Costello, "Smart projectile state estimation using evidence theory," AIAA Atmospheric Flight Mechanics Conference, Aug. 2011.

[87] H.-R. Bae, R. V. Grandhi, and R. A. Canfield, "Uncertainty quantification of structural response using evidence theory," *AIAA Journal*, vol. 41, pp. 2062–2068, 10 2003.

[88] T. Bayes and R. Price, "An essay towards solving a problem in the doctrine of chance. by the late rev. mr. bayes, communicated by mr. price, in a letter to john canton, a. m. f. r. s.," *Philosophical Transactions of the Royal Society of London*, pp. 370–418, 53 1763.

[89] J. BoYang and D. LingXu, "Evidential reasoning rule for evidence combination," *Artificial Intelligence*, vol. 205, Sep. 2013.

[90] J. Heendeni, K. Premaratne, M. Murthi, J. Uscinski, and M. Scheutz, "A generalization of bayesian inference in the dempster-shafer belief theoretic framework," International Conference on Information Fusion, Jul. 2016.

[91] P. Shenoy, "Conditional independence in valuation-based systems," *Journal of Approximate Reasoning*, pp. 203–234, 10 1994.

[92] H. Xu and P. Smets, "Evidential reasoning with conditional belief functions," 10th Conference on Uncertainty in Artificial Intelligence, 1994, pp. 598–605.

[93] C. SIMON, P. WEBER, and E. LEVRAT, "Bayesian networks and evidence theory to model complex systems reliability," *JOURNAL OF COMPUTERS*, vol. 2, 1 Feb. 2007.

[94] V. Nguyen, "Approximate evidential reasoning using local conditioning and conditional belief functions," Sidney, Australia: Conference on Uncertainty in Artificial Intelligence, Aug. 2017.

[95] E. Pollard and B. Pannetier, "Bayesian networks vs. evidential networks: An application to convoy detection," Communications in Computer and Information Science, Jun. 2010.

[96] A. Benavoli, B. Risti, A. Farina, M. Oxenham, and L. Chisci, "An application of evidential networks to threat assessment," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 45, pp. 620 –639, May 2009.

[97] M. Rodriguez and J. Geldart, "An evidential path logic for multi-relational networks," *AAAI Spring Symposium - Technical Report*, Oct. 2008.

[98] L. Shastri and J. Feldman, "Evidential reasoning in semantic networks: A formal theory.," 9th international joint conference on Artificial intelligence, Jan. 1985, pp. 465–474.

[99] A. K. Barrette, "Artificial intelligence techniques to the non-cooperative identification (ncid) problem," AIAA Computers in Aerospace VII, Oct. 1989.

[100] NumPy.org, *Numpy*, `https://numpy.org/`, Web Page, 2019.

[101] R. R. Yager, "On the dempster-shafer framework and new combination rules," *Information Science*, vol. 41, no. 2, 1987.

[102] K. J. Astrom, *Introduction to Stochastic Control Theory*. New York Academic Press, 1970, pp. 7–12, 278–293.

[103] *Tesla*, `https://www.telsa.com/`, Web Page, 2020.

[104] *Waymo*, `https://www.waymo.com/`, Web Page, 2020.

[105] Z. D. R. Pan Y. Peng, "Belief update in bayesian networks using uncertain evidence," IEEE International Conference on Tools with Artificial Intelligence, Nov. 2006.

[106] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters*, vol. 20, no. 8, 861–874, 2006.

[107] P. Guttorp and T. L. Thorarinsdottir, "What happened to discrete chaos, the quenouille process, and the sharp markov property? some history of stochastic point processes," *International Statistical Review*, vol. 80, pp. 253–268, 2012.

[108] L. VELOS ROTORS, *Velos rotors*, `https://velosuav.com/`, Web Page, 2020.

[109] EMLID, *Emlid*, `https://emlid.com/`, Web Page, 2020.

[110] R. Penrose, "A generalized inverse for matrices," 3, vol. 51, Proceedings of the Cambridge Philosophical Society, Cambridge University Press, Jul. 1955.

VITA

Joel Dunham received his B.S. degree in Aerospace Engineering from Iowa State University in 2006, an M.S. degree in Aerospace Engineering from the Georgia Institute of Technology in 2008, and an M.B.A. from the University of Texas at Austin in 2012. He has over 10 years of industry experience and is finishing his Ph.D. in Aerospace Engineering at the Georgia Institute of Technology while working as a research engineer at the Georgia Tech Research Institute. His areas of interest are widespread and include Decision Theory, Unmanned Systems, Risk Analysis, Information Fusion, and Simulation with an emphasis on applications of these technologies.