2019 International Conference on Unmanned Aircraft Systems (ICUAS)
Atlanta, GA, USA, June 11-14, 2019

Safe decision making for risk mitigation of UAS

Lina Castano, and Huan Xu, *Member, IEEE*

***Abstract*—This work entails the development of a theoretical framework for the fast and safe reaction of Unmanned Aerial Systems to flight anomalies. The proposed framework uses behavior trees to design behaviors that have safety properties and uses vehicle states to determine best risk control response. A case study is presented where a fixed-wing UAV experiences four types of hazardous scenarios which consist of different types of aircraft faults and external obstacles. The UAV safely reacts by avoiding obstacles when possible, or mitigates the faults by either finding a nearby emergency landing location, or choosing a safe ground impact point far from populated areas. These behaviors have the potential to be used across different aerial robotics platforms, to increase safety amid human error, environmental hazards and aircraft failures.**

***Index Terms*—Safe UAV control, risk mitigation, safe response, UAS safety, behavior trees**

I. Introduction

SAFETY is of paramount importance for successful integration of Unmanned Aerial Systems (UAS) into the National Air Space (NAS). UAS aircraft operations are seeing a rapid growth due to their customizability and flexibility, and can perform tasks typically undertaken by manned aircraft or that may pose a risk to humans. These tasks span a variety of applications, from aerial photography and infrastructure inspection [1], to disaster relief [2], mining [3] and geographical mapping and exploration [4]. Persistent surveillance, communications relays, and combat search and rescue are some of the main market drivers for UAS technology development [5]. The global UAS payload market alone is expected to reach 68.6 billion by 2022 [6], with a compound growth rate (CAGR) of 4.6%. Therefore, functional integration of UAS into the NAS will be a catalyst for widespread UAS operations and economic growth. UAS platforms need to combine features that allow for modular architectures in order to achieve diverse types of missions.

Integration into the NAS involves UAS operating side by side with manned aircraft, sharing the same airspace and using many of the same air traffic management (ATM) systems and procedures [5]. The Federal Aviation Administration (FAA) has taken two important steps towards this integration addressing small UAS (sUAS)- those under 55 pounds. The first one entails the Registration and Marking requirements, and the second one, corresponds to Title 14 part 107 of the Code of Federal Regulations (CFR). The latter one allows for small UAS operations to be conducted routinely within the visual line of sight (VLOS). A comprehensive set of additional regulations for sUAS were provided by the FAA's Advisory Circular (AC) no. 107-2 [7]. Beyond visual line of sight (BVLOS) as well as fully autonomous operation of UAS have not been regulated as they require that major technological and operational challenges be addressed first.

One of the key challenges for integrating UAS into the NAS is the development of decision-making software to mitigate potential navigation hazards, such as midair collisions (MACs) or near mid air collisions (NMACs) with other aircraft or other obstacles [8]. This requires sense/detect and avoid technologies and associated software to ensure that the UAS maintains a safe distance from other aircraft, whether they are cooperative or uncooperative. Separation procedures for commercial passenger and military aircraft are usually conducted by Air Traffic Control (ATC) operators along with pilot's visual flight rules (VFR) or instrument flight rules (IFR), depending on the weather and visibility conditions [9]. However, for sUAS, separation procedures typically require that the pilot in command (PIC) visually detect, recognize, and predict a UAS potential collision or system fault. This places all safety-related decisions on the PIC, who may not be able to react fast enough to fast moving obstacles or sudden system faults [10]. System faults can also pose serious safety hazards [11]. Other fault types include aircraft faults, operational faults, and complex scenarios which may lead to emergency procedures.

Approximately 70% to 80% of all manned aviation accidents are caused by pilot error [12]. Accident rates of unmanned aircraft are much higher than that of manned aircraft. Among these incidents, the largest percentage of accidents involving human error is attributed to unsafe acts [11]. Within this classification, decision errors accounted for the highest percentage of human error accidents, followed by skill-based errors and perceptual errors. Decision errors include following an improper procedure, misdiagnosing an emergency, having the wrong response to an emergency, or inappropriate maneuver [12]. Skilled-based errors are those caused by a breakdown in visual scan, failure to prioritize attention, omitted steps in procedure, among other factors. Perceptual errors include spatial disorientation and failure to judge distance, altitude or airspeed. Furthermore, human error accidents were mainly due to individual failures, followed by standards failures, leader failures, training failures, and support failures [13]. Skilled based errors and perceptual errors can propagate to produce decision errors [14].

It is clear that decision-making plays a central role in safety of unmanned aviation, either to assist a PIC or to operate fully

This work was supported in part by the Maryland Industrial Partnerships (MIPS) Program and by Millennium Engineering and Integration, Co.

L. Castano is with the Aerospace Engineering Department, University of Maryland, College Park, MD 20742 USA e-mail: linacs@umd.edu.

H. Xu is with the Aerospace Engineering Department and Institute for Systems Research, University of Maryland, College Park, MD 20742 USA e-mail: mumu@umd.edu.

978-1-7281-0332-7/19/$31.00 ©2019 IEEE

autonomously within the constraints of current regulations. In addition, as reported by [11], electrical and mechanical reliability play as much of a role in UAS accidents as human error. Human factors issues are dependent on the particular UAS being flown, the type of automation and the user interface and command interface to the system. Therefore, an efficient decision-making software will need to not only have varying degrees of automation and human-interaction [15], but also monitor for faults and have a safe and fast response to hazardous circumstances that the UAS may experience.

This work addresses the need for decision-making software for safety of UAS. Our approach consists of generating UAS behaviors, or sets of sequences of actions, that can be implemented on higher level software routines. These responsive behaviors will allow for fast reaction to flight anomalies [16], which include encountering static or dynamic obstacles and different types of aircraft and system faults. In the proposed framework, the decision layer has access to all the relevant vehicle and system states and after identification of the hazard, decides to activate the safest response which may include one or a combination of the following: fault tolerant operation [17], obstacle avoidance maneuvers [18], landing strip determination and emergency landing [19], flight termination to a crash point [20] or parachute deployment [21]. This work is constructed in the Behavior Trees framework, which is a behavior design tool from game artificial intelligence (AI). Its scalability, reusability and modularity make it an ideal tool for formulation of UAS decisional behaviors [38].

This paper is structured as follows: Section II will provide an overview and definitions for aeronautical decision-making as well as an overview of autonomous decision making algorithms, Section III will provide the decision-making behavior tree formulation for UAS, Section IV will provide case studies and discussion, and Section V the conclusions.

II. Theoretical Framework

This section will provide the theoretical background of the pilot's decision-making process. It will then provide an overview of relevant work in the field of autonomous agents. The core concepts of behavior trees will be presented in section IIIA.

A. Aeronautical Decision Making

Aeronautical decision making (ADM) is a systematic approach used by pilots to determine the best course of action in response to a given set of circumstances [14]. Safe decision-making requires the pilot to have full situational awareness at all stages of flight. Situational awareness is the accurate perception and understanding of all the factors involved in the fundamental risk elements, as shown in Fig. 1. The fundamental risk elements consist of hazards associated to:

- Pilot: Decision-making, judgment and evaluation.
- Aircraft: Physical component faults.
- Environment: Unexpected weather and obstacles.
- Operation: Procedure, mission plan.

All these elements should effectively combine to provide situational awareness. A pilot's decision process can

Figure 1. Fundamental risk elements for situational awareness: core elements which affect flight safety at all stages of flight.

be described by a classical linear model named DECIDE, shown in Fig. 2, which is an acronym for Detect, Estimate, Choose, Identify, Do and Evaluate [14]. This analytical model advocates sequential consideration of alternative decisions and possible outcomes to respond to a changing situation. Other mnemonic-based methods include the 3P (Perceive, Process, Perform) model [22], FOR-DEC (Facts, Options, Risks and Benefits, Decision, Execution, Check), and SOAR (Situation, Options, Act, Repeat) [23], among others. These methods entail generation of an exhaustive list of options, evaluation and assessment of the relative value of each option and then implementation of the option with the most desirable outcome. It has been observed that the more experience pilots have the more automatic the retrieval of relevant courses of action from memory.

For the experienced pilot, there is a continuous updating of the current situation with direct retrieval of the appropriate course of action. Therefore experience turns structured and sequential decision procedures into direct retrieval of the correct decision for a particular scenario [24], to minimize reaction time in emergency situations. Aviation decision making, especially in situations with unexpected changes and crises, requires a skilled use of both deductive and inductive reasoning. Expert pilots have the ability to find meaningful patterns which yield a behavioral resultant based upon intuition. The intuitive pilot monitors, anticipates and considers the need for action in advance of the moment it is required.

Experience-based models allow for decision-making in naturalist decision environments [25]. Naturalistic decision-making (NDM) models focus on the aspect of finding a solution quickly if a given criterion is satisfied. Therefore if an acceptable course of action is found the search ends and is executed by the decision-maker. One model which allows for rapid decision making is the Recognition-Primed (RPD) decision model [26]. In this recognitional model, experience translates into automatic decision making. Essential characteristics of naturalistic decision environments include: time pressure, dynamic scenarios, high risk, shifting goals, ambiguous data, and cue learning [25]. It is important to note that even experienced pilots may not be able to react fast enough to changes in flight conditions. Performance capabilities of many aircraft, in speed and rates of climb or descent, may result in high closure rates limiting the time available for detection, information processing, decision-making and response action [10]. This is particularly important with sUAS, as they possess higher bandwidth dynamics.

Decision-making, however, is more than picking a course of action; decision strategies need to work in operational contexts and are situation dependent. Intuitive or non-analytical situation assessment is a crucial part of the decision process. Real-world decision making requires decisions to be made very efficiently (automatic) and in the most methodical ways possible (analytical). Based on outcomes of experienced pilots versus novices, it is apparent that there are safer behaviors among pilots according to their experience level, in addition to the available information for complete situational awareness.

This paper focuses on modeling the pragmatic aspects of safe pilot behavior and applies it to UAS. High level controllers can be set to emulate safe reactive behaviors for rapid yet structured mitigation of flight hazards.

1) Safety Elements for Decision-Making: The constitutive components of pilot decision-making for safety of aircraft include:

- Situational Awareness (SA)
- Risk assessment and management (RA&RM)
- Task management (TM)
- Information management (IM)
- Automation management (AM)

As was mentioned in the previous section, situational awareness is the basis of decision making. It provides the decision-maker with all the information required for mitigation of hazards. Safety risk assessment is another important component, it entails identification of safety related hazards and subsequent mitigation of associated risks. This process involves reducing unnecessary risks when possible, making risk decisions at the appropriate level according to the situation at hand, and applying risk management at all stages

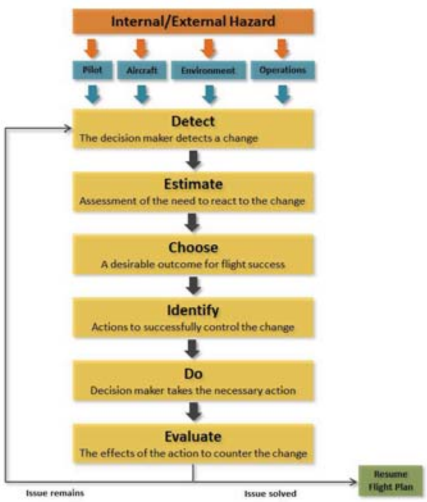

Figure 2. Classical model for aeronautical decision making

of the flight [14]. Task management consists of properly managing concurrent tasks, and information management is the process which pilots use to collect and prioritize information from all available sources. Automation management entails the ability to correctly manage the aircraft's automated systems such as autopilots and flight management systems.

Ensuring safety of UAS is a key component towards certification and integrated operation in the NAS. FAA certification systems require substantiation on identification of cause and effect of hazards, risk assessment and validation/verification of safety requirements [27]. Safety certification requirements largely depend on operational requirements, which include UAS performance attributes and design.

In this work, we propose development of reactive behaviors for UAS, that comply with UAS safety requirements. These reactive decision-making behaviors can be implemented in software and interfaced with corresponding embedded systems for safe control of UAS.

B. Decision Making for Autonomous UAS

Autonomy is achieved by introducing a decisional layer in the UAS architecture. An autonomous UAS is able to make independent decisions in uncertain, nondeterministic and partially observable environments. It requires methods that generalize to unpredictable scenarios and reason promptly to reach human level reliability and react safely to complex environments. Fully autonomous platforms are not yet available, however incremental progress is being made by introducing degrees of automation. SAE defines levels of autonomy as ranging from no-autonomy or level 0, to fully autonomous, or level 5 [28]. These levels go according to the specific performance that an autonomous system has with respect to a human operator, as it relates to a dynamic task of interest.

A general layered architecture for autonomy is presented in Fig. 3, based on [28]. In this figure, an autonomous system is modeled as a layered assembly consisting of the following: a decision layer, an executive layer, functional layer, and a hardware layer. We have added a fourth layer which interacts with all of the other system components, as there will be safety metrics at each of these levels. Layered architectures are the most versatile agent types and often include the layers or blocks of reactive, logic-based and coordinated behaviors.

Decision methods for autonomous agents can be classified as one of the following: Planning or deliberative [29], Behavior-based, belief-desire based, situation calculus based, layered methods [30], and hybrid blends of the methods [29]. All of these methods contain elements of logic-based deduction rules and goals, as well as reactive components. Reactive decision making usually entails logic with fast rules, behavioral embodied intelligence, and finite state machines. One of the best known behavior based reactive agent architectures is the subsumption architecture[31]. In this architecture each behavior is represented in a separate layer, with direct access to sensory information. These layers have a hierarchical arrangement where they can be absorbed or inhibited by higher level ones.

In the literature, other approaches to computational intelligence and autonomous robot controller design have been explored through several different algorithms. One of the approaches is found in biologically inspired robotics such as evolutionary robotics, which seeks to obtain self-adaptation and self-learning capabilities. Genetic algorithms and genetic programming have been used for generating robotic behaviors [32]. Controllers derived by evolutionary methods have advantages in fast adaptation time [33]. Other methods include neural networks as the basic blocks for the control system [34], and fuzzy logic. Fuzzy logic allows for modeling the nonlinear correspondence between input information and control output. It incorporates heuristic control knowledge in the form of if-then rules, and is a suitable alternative when the system cannot be fully modeled. Fuzzy controllers have a good degree of robustness amid large variability and uncertainty in the control parameters [35]. There are also combinations of these methods, such as the fuzzy-genetic system, which is a typical evolution mechanism in evolving adaptive robot controller. Some of these methods however have different drawbacks. Neural networks for instance may require abundant training time, and genetic algorithms are computationally expensive.

Other methods include reinforcement learning, where the autonomous agent learns a behavior through trial and error interactions with a dynamic environment. Partially observable Markov decision processes provide a framework for agents that can learn how to act, and incorporates uncertainties about an agent's perceptions, actions and feedback. Another approach is found in finite state machines and hybrid dynamical systems, where each state and set of transitions represent some desired behavior. State machines can become difficult to define and represent for complex environments.

Behavior trees (BTs) are similar to a combination of hierarchical finite state machines or hierarchical task network planners. BTs allow for a more intuitive and less complex formulation of decision processes, similar to human intelligence. Furthermore, Behavior Trees generalize Decision Trees (DTs), which have proved to be valuable tools for the classification, description and generalization of data and decision processes [36]. Behavior Trees generalize Decision Trees in that these allow for nodes to output information, as in DTs no information flows out of action nodes, which makes failure scenarios difficult to handle.

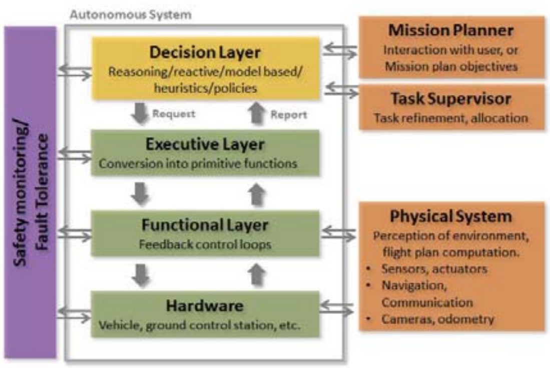

Figure 3. Architecture for decisional autonomy with safety monitoring and fault tolerance.

Our work is based on the formulation for behavior trees, which will be described in the next section.

III. Safe Decision Making for UAS

UAS are high bandwidth systems, whereby rapid changes or anomalies require fast identification and response. The fault detection process and response can be conducted using behavior trees in a modular way. Section IIIA will introduce the core concepts of Behavior Trees, and section IIIB will present our formulation of BTs for safety and risk mitigation of UAS in the presence of flight anomalies.

A. Behavior Trees

A Behavior Tree is a graphical modeling language and representation of 'behaviors' or execution of actions which are based on observations in a system [32]. These have been used for defining character behaviors in computer games Artificial Intelligence (AI) [37]. Behavior trees make hybrid dynamical system transitions implicit, therefore simplifying the encoding of state transitions. In addition, BTs are modular and allow for reactive behavioral structures to be created intuitively. Actions, conditions and sub-trees can easily be added or removed from a behavior tree without having to make careful revisions of all the state transitions [38]. In addition, BTs for control of autonomous agents enable prioritization of behaviors or responses. Therefore, this is an ideal framework for creating behaviors for safe reactions to hazards encountered by a system, in particular a UAS.

A behavior tree has four nominal node types: *selector, sequence, parallel*, and *decorator*. Leaves can be either an *action* or *condition* node. Selectors and sequences are composite nodes. When prompted for its status, each node can return: *success, failure*, or *running*. Fig. 4 depicts the control structure of a BT, where boxes represent nodes and lines represent edges. If two nodes are connected by an edge, the outgoing node will be called the parent node and the incoming one a child node. Nodes that have no children are called leaves, and the node without parents is called the root node. In this figure, a selector and a sequence control structure are illustrated.

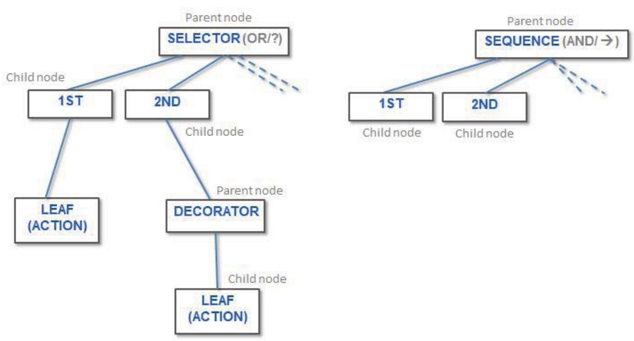

Figure 4. Selector and sequence node control structures in behavior trees.

A selector will ‘succeed’ if one of its children succeed and ‘fail’ if all of its children fail. It will return ‘running’ if one child returns ‘running’. A sequence node will succeed if all its children succeed and fail if one child fails. Selectors can be thought of OR operators and sequences as AND operators; however with the added ‘running’ status which extends the Boolean capabilities. In addition it allows for goal directed execution of one task after the other using sequence nodes and allows for a series of fallback actions via selector nodes.

All operational aspects of UAS can be formulated in terms of combinations of sequences and selectors. A flight plan and mission waypoints can be formulated as a sequence behavior tree. This tree will encode the phase flight sequence and any other mission goals. Each sub-node in the sequence can also contain a sequence of steps that correspond to normal operation of the UAV. Each subnode, Taxiing, Take-off, cruise, waypoint navigation, Landing, will also have a sequence of steps, or sequence sub-tree.

B. Behavior Trees for Fault Detection and Mitigation

Behavior trees (BTs) can be used in the detection of faults which evolve into failures and/or malfunctions. Due to their reactive functionality, behavior trees are great protocols for specifying safety features of systems, and systems of systems. A behavior is often composed of a sequence of sub-behaviors that are task independent. The modularity properties of BTs are useful in that it not only allows for formulating behaviors at all levels of the operation of an autonomous system, but also allows for changing actions within branches without having to re-plan the entire task and mission. Actions can be added and extended to any branches and sub-branches of a given behavior. This is possible due to the two way control transfers used by behavior trees, as compared to the one way control transfers that finite state machines use. Reusability of formulations are another very useful aspect of behavior trees, and provides additional insights into how systems should effectively react to anomalies.

1) Detection Sub-Trees: The detection process entails precursor detection, fault detection, classification, and verification, as shown in Fig. 5. After detection, a mitigation strategy is chosen as well as a decision on whether to avoid obstacles, or land the aircraft, or perhaps choosing a crashing location, depending on a number of external as well as internal factors.

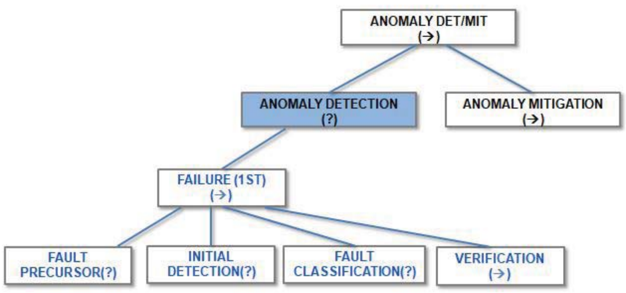

Figure 5. Anomaly detection and mitigation behavior tree with expanded anomaly detection sub-tree.

A sequence behavior tree for fault detection and mitigation can be formulated as shown in Fig. 5. This allows for rapidly going back to the root node in case a false positive is issued. The critical failure evaluation will also stop if it can’t verify that there has been a critical failure of a specific type. This figure shows an expanded view of the initial fault detection sequence node. It first goes into a selector node which determines whether the fault is found in the sensors, actuators or if it is a path deviation that is causing the anomaly. When it goes out of this subnode, it continues on to initial detection, classification, verification, and if the fault is verified, the tree jumps to the mitigation branch, which is also comprised of sub-nodes depending on the desired action or action sets. Terminal actions can be defined using decorator nodes and leaves.

A schematic of the high level composition of the detection part of the behavior tree can be formulated as shown in Fig. 6. Notice that the system doesn’t need to execute all the sub-branches in each type of failure initially, this will occur once a precursor is evaluated as positive. This selective evaluation allows for faster detection of faults.

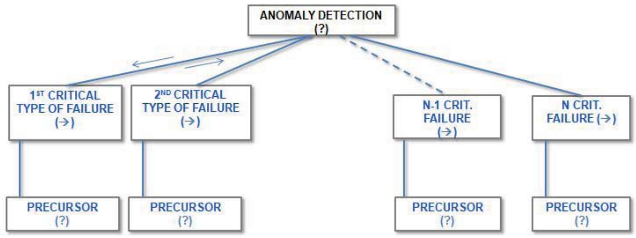

Figure 6. Anomaly detection top level tree.

An anomaly detection tree based on most critical flight anomalies for autonomous UAS is seen in Fig. 7. The most critical faults in this formulation correspond to engine failure and loss of link, which is the main communication avenue with the ground station. Different priority checks are then applied based on critical loss of link failure possibilities, based on statistical observations and most common events. Another anomaly detection sub-tree is described in Fig. 8, where the most important critical checks involve fuel gauge and altitude loss, for engine failure detection.

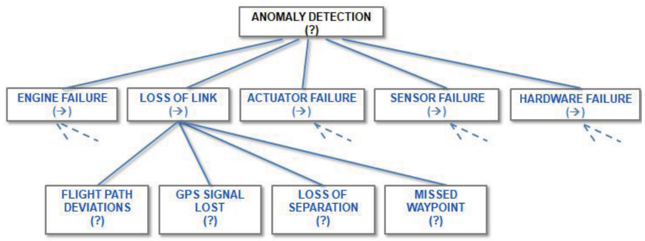

Figure 7. Anomaly detection tree based on most critical flight anomalies for UAS, and expanded loss of link sub-tree.

A number of fault detection and diagnosis methods can be used under the BT framework. Both model-based and data-based methods can be used. Model based methods for

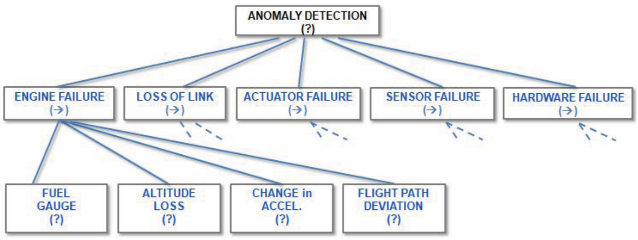

Figure 8. Anomaly detection tree based on most critical flight anomalies for UAS, and expanded engine failure sub-tree.

detection and diagnosis include regression analysis, Kalman filters and observers. Data based methods include statistical classifiers and neural networks.

2) Mitigation Sub-Trees: Fig. 9 shows the same root tree as in Fig. 5, with expanded anomaly mitigation subtree. In this subtree, the system first explores different mitigation strategies and then makes an assessment of the resulting stability and controllability of the aircraft. If none of the strategies are successful, the tree drives the UAS into an emergency procedure. This procedure may entail obstacle avoidance, landing, etc.

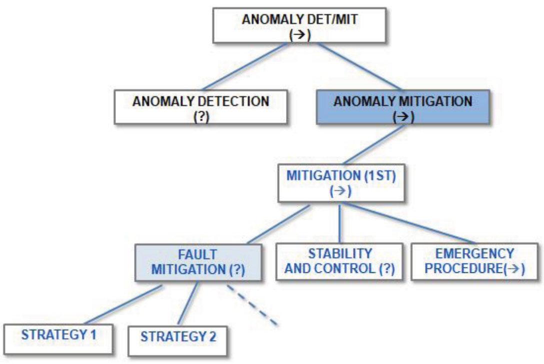

Figure 9. Anomaly detection and mitigation behavior tree with expanded anomaly mitigation sub-tree, and mitigation strategies sub-subtree.

One of the advantages of using BTs is that they can be adapted and inserted into other tree structures. Therefore these fault detection and mitigation trees can be inserted into any branch of a nominal flight plan.

Using the modularity aspect of the BT framework, we now define one more root level tree which corresponds to a hazard identification and mitigation node, as shown in Fig. 10. This top level tree includes obstacle detection and mitigation, as well as environmental hazards and mitigation. The first branch of this mode corresponds to the anomaly detection and mitigation tree, shown in Fig. 5 and Fig. 9. Mitigation strategies are applied in increasing levels of severity depending on the type of fault. Prioritization of mitigation strategies are set according to the type of fault. The sub-tree nodes in Fig. 10 should include all the risk elements described in Fig. 1, including the pilot's human factors, in the case of collaborative UAS operations.

Mitigation strategies include reconfigurable control techniques such as LQR, model predictive control, gain scheduling, adaptive control, as well as neural networks and machine learning techniques. If mitigation strategies are not sufficient for safe operation to the next mission waypoint, a landing location is selected. Previous work by authors demonstrated the utility of a fast reaction to a flight anomaly by selecting an emergency landing location [19]. Another aspect of safe reactive behaviors of autonomous systems consist of using population data and geographical databases for choosing a new landing location, or in the worst case scenario, choosing a location for crashing, in the case of an autonomous air vehicle. In our previous work [20], this was demonstrated by a fast resolution to an unrecoverable fault where a crashing location was selected in a low population geographical area. Our work also demonstrated the use of a fast collision avoidance algorithm to mitigate the chance for mid air collisions [18].

Behavior trees can therefore be defined through logic, experiential information, and data-driven or statistical approaches. BTs can also be generated automatically by machine learning methods or genetic algorithms [32]. It is also important to note that safety properties can be included as detection/mitigation behaviors, as discussed in this section, but can also be included at every point in an autonomous UAS architecture (Fig. 3). Furthermore, behavior trees can be used with tools from formal verification such as model checking, to verify a system for safety requirements [39].

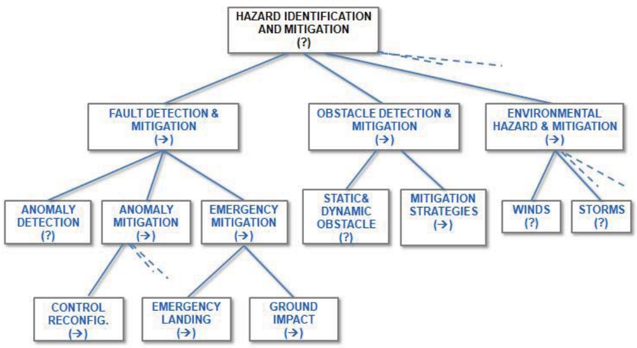

Figure 10. Hazard identification/mitigation BT, and expanded sub-trees for fault detection/mitigation, obstacle detection/mitigation.

C. Concurrent execution

Behavior trees allow for intuitive encoding of aeronautical decision making, both in the sequential or classical approach (Fig. 2) by going through all the possible mitigation solutions of a flight anomaly (Fig. 9), and a rapid response to a given detected anomaly, by executing first mitigation strategies which are the most likely effective methods. The remaining options are not executed if the first mitigation strategy is completed. Therefore, this models experiential behavioral reactivity to aircraft faults and navigational anomalies such as encountering obstacles along the flight path or bad weather. This also allows for faster and more effective reaction to faults, which is especially useful for lighter UAS which may not have access to powerful computing resources.

If use of plenty of computational power is feasible, the more robust option is to execute detection trees in parallel, such that secondary concurrent events are not ignored while the system mitigates primary ones. In this case, fault detection/mitigation, obstacle detection/mitigation and environmental hazard detection/mitigation (Fig. 10), would run in synchronized parallel computing units. Additional logic would then be required to sort system action in cases where multiple faults and anomalies are occurring simultaneously.

IV. Case Study

A. Simulation Setup

Simulations were conducted using Simulink, and consist of a UAV model, aircraft controllers, path planning and navigation, and hazard mitigation modules. These are shown in Fig. 11 with top level data connectors. In the simulation, a UAV is presented with a set of waypoints or mission plan over a specified geographical area. During the flight, it encounters different types of hazards which it tackles using the Behavior Tree logic and hazard mitigation behaviors.

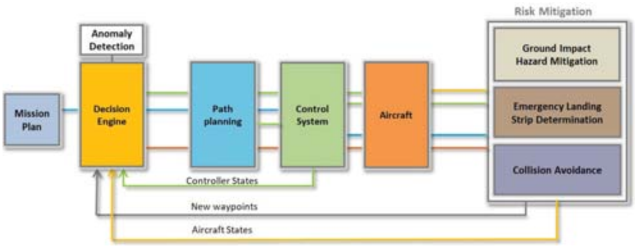

Figure 11. Software simulation: Decision-making, flight dynamics, path planning, controllers and risk mitigation modules.

1) UAV Flight Software: The UAV flight software consists of the path planning, control system and aircraft modules, as seen in Fig. 11. The path planning module consists of a non-linear guidance law and issues heading, altitude and speed commands to the low level controller block, or control system. This module contains a speed and altitude autopilot, as well as PID controls for roll, pitch and yaw. Aileron, throttle, elevator and rudder commands are then sent to the aircraft model. The aircraft consists of a 6-DOF UAV, modeled with standard linearized equations of motion. The linearized equations make use of stability derivatives for calculation of forces and moments about the center of gravity. Table I shows a set of characteristics of the general aviation UAV used in the simulation.

Table I. General Aviation UAV characteristics used in simulation.

Variable	Definition	Value
$airfoil$	wing airfoil	SD7037
C_{Li}, C_{Di}, C_{Mi}	Lift, Drag, Moment coeffs. derivs.	from airfoil
w_c	wing aerodynamic chord	0.22 m
L_w	wingspan	1.4 m
T_w	wing taper	1.95
AR	wing aspect ratio	6.4
v	aircraft velocity	20 m/s
α_{max}	max angle of attack	6.25°
m	maximum take off weight	1.2 kg

2) Flight Hazards: For this work two fault modes were introduced in the aircraft module.

1) Engine failure - UAV engine does not produce thrust but can still maneuver in roll, pitch and yaw.
2) Engine and rudder failure - UAV engine does not produce thrust. Rudder is stuck near trim. Aircraft can still maneuver with remaining control surfaces.

These faults have a sudden onset and are permanent for the remainder of the flight. In both of these cases, the general aircraft equations reduce to glider equations, due to removing the thrust or engine force. All of these effects are taken into account in the aircraft model and associated controllers.

Static and dynamic obstacles are also introduced such that they cross the UAV's intended path, as initially set by the mission plan waypoints.

3) Decision Making Module: The decision engine has access to all state variables in each of the modules and can activate the risk mitigation modules depending on the hazard that has been detected. The risk mitigation modules in our current simulation consist of Ground Impact Hazard Mitigation (GHIM) [20], emergency Landing Zone Determination and Path Planning (LDP) [19], and Collision Avoidance (FGA) [18]. The decision making logic will first assess whether there is a hazard, then it will determine the type of hazard, and finally it will activate a mitigation sequence based on the type of aircraft fault detected, or the type of obstacle detected. In the case of the fault detected, we have set the mitigation priority to be control reallocation, followed by landing strip determination and followed by GHIM in the case that none of the previous mitigation strategies have been successful.

Fig. 12 shows the Simulink stateflow chart of the Detection/Mitigation Behavior Tree. In this tree, the detection loop is the top most layer and the mitigation strategies sub-trees are expanded at each detection block. The detection blocks correspond to fault mode detection, static obstacle detection and dynamic obstacle detection. The last block is a control block to close the detection loop. The structure of these trees follow a similar behavior to the one shown in Fig. 9.

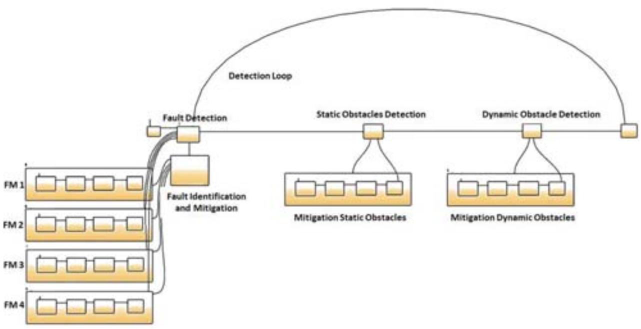

Figure 12. Decision Engine in Simulink: top level detection tree and expanded mitigation sub-trees.

The decision-making BT communicates with the other modules via external variables and uses triggers to activate the appropriate hazard response. Finding a landing strip as an emergency procedure entails the use of digital elevation models, terrain ruggedness index and land cover index data. It also is determined by proximity and quality of the landing strip. Finding a ground impact point in the event of a critical flight emergency, entails the use of a population dataset and the aircraft model features to generate a feasible footprint. Obstacle avoidance is conducted by using a geometric algorithm assuming perfect sensory information from devices such as ADS-B technology and LIDAR. This algorithm uses distance to calculate a critical avoidance time and resultant avoidance maneuver.

4) Case studies: In this work we have considered the following scenarios that present different types of collective hazards for a UAV.

1) Case study 1: UAV encounters one static obstacle and two dynamic obstacles along its originally intended path. The aircraft is able to reach its original last mission waypoint.
2) Case study 2: The UAV enters an engine fault mode, and it also encounters two dynamic obstacles along its path. The aircraft finds a crash point in an area with low population density, after unsuccessful search for a landing strip.
3) Case study 3: The UAV enters an engine malfunction and has a stuck rudder, and it also encounters one dynamic obstacle along its path. The aircraft finds a crash point in an area with low population density, after unsuccessful search for a landing strip.
4) Case study 4: The UAV enters an engine fault mode, and it also encounters one static obstacle along its path. The aircraft finds a suitable landing zone and generates additional emergency landing points to guide the UAV towards the landing strip.

B. Case study 1: Static and Dynamic Obstacles

In this scenario, the UAV encounters one static obstacle and two dynamic obstacles along its waypoint course. The decision making BT is able to detect and trigger a corresponding avoidance maneuver. The aircraft is able to finish its intended mission plan and reaches the last mission waypoint in a total mission duration of about 15 minutes. Fig. 13 shows a satelite image of the mission and UAV trajectory. The red dashed lines represent the trajectory of the dynamic obstacles.

Figure 13. UAV waypoint navigation: Static and dynamic obstacles are mitigated by Decision Making BT.

C. Case study 2: Engine Failure and Dynamic Obstacles

In this case study, we introduce an engine fault at close to 5 minutes of flight, and two dynamic obstacles along its path. The UAV will detect each anomaly and will try to mitigate it using its decision-making BT. We have made the assumption that the control reallocation mitigation has not been successful, and that the main mitigation strategies correspond to emergency landing (LDP), emergency ground impact (GHIM), and obstacle avoidance (FGA). Fig. 14 shows an earth view of the mission plan, dynamic obstacles, and an outline of the reactive decision process.

The UAV's decision process produced by the BT in this case scenario is described as:

1) Beginning of mission plan, traveling towards WP1 from home location.
2) BT detects moving obstacle 1, detection node.
3) BT mitigates moving obstacle 1 - activates FGA, mitigation node.
4) UAV resumes flight towards now WP1.
5) BT detects moving obstacle 2, detection node.
6) BT mitigates moving obstacle 1 - activates FGA, mitigation node.
7) UAV resumes flight towards WP2.
8) BT detects Engine Fault, detection node, FM1 detection.
9) BT mitigates Engine Fault with control reallocation, but doesn't work, so it chooses to look for an emergency Landing strip.
10) BT mitigates Engine Fault with LDP but doesn't work, so it then chooses a crashing point for flight termination.
11) BT drives Aircraft to safe crash point generated by GHIM, in a non-populated area.

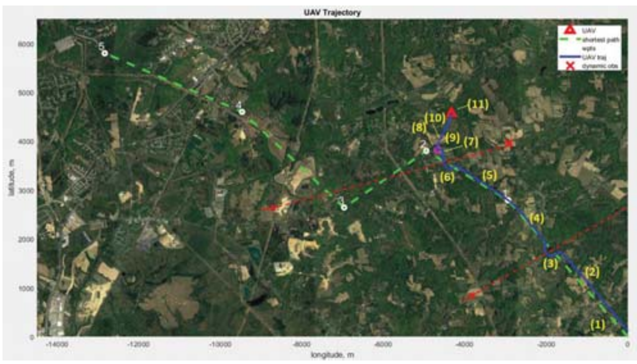

Figure 14. UAV engine failure mitigation and obstacle avoidance, using Decision Making BT.

Fig. 15 shows the resulting 2D path of this case scenario over a population density map based on Landscan data. The UAV crashes on a region with low population density, as per developed by the GHIM module. Fig. 16 shows the Euler angles and Euler rates of the UAV during its flight and hazard mitigation maneuvers.

D. Case study 3: Engine and Rudder Failure, and Dynamic Obstacle

In this scenario, the aircraft starts out its mission plan with no actuator failures. It first avoids a dynamic obstacle, i.e. a

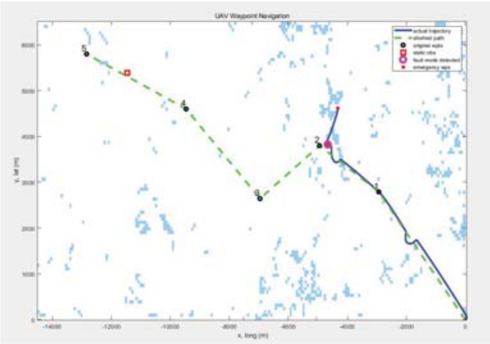

Figure 15. Population density map and UAV trajectory: UAV engine failure mitigation and obstacle avoidance, using Decision Making BT.

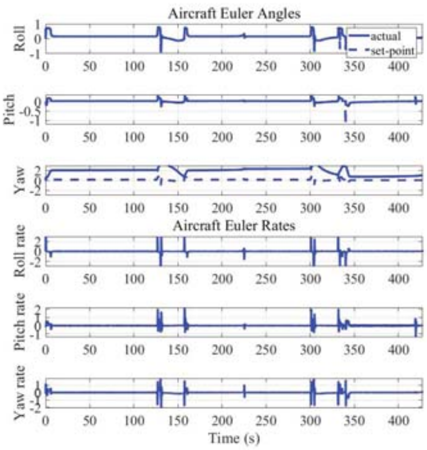

Figure 16. Euler angles and euler rates: UAV engine failure mitigation and obstacle avoidance, using Decision Making BT.

natural flier or another aircraft, that crosses its intended path. After this point, the UAV enters an engine malfunction and looses thrust. In addition, its rudder gets stuck near trim. At this point, the BT searches for an emergency landing location, but then decides to instead find a crash point after not finding a suitable landing strip. The BT then drives the aircraft to a ground impact location to terminate its flight.

Fig. 17 shows a close up image of the UAV trajectory. In this figure, the UAV is driven to an unpopulated area for flight termination.

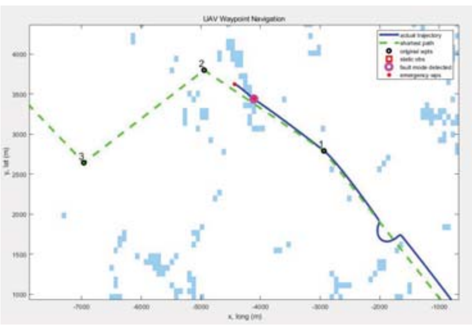

Figure 17. Population density map and UAV trajectory: UAV engine and rudder failure mitigation and dynamic obstacle avoidance, using Decision Making BT. A ground impact point is selected using GHIM.

E. Case study 4: Engine Failure and Static Obstacle

For this case, a different mission waypoint course is chosen. During its flight, the UAV encounters one static obstacle along its path and enters an engine fault mode shortly after. In this case, the aircraft finds a suitable landing zone and generates additional emergency landing points via a bidirectional rapidly exploring random tree (BiRRT) path planning to guide the UAV towards the identified suitable landing strip. Fig. 18 shows the UAV trajectory, static obstacle avoidance, fault detection and emergency landing. Fig. 19 shows an expanded view of the emergency landing waypoints generated by the LDP module, the trajectory and the location of the landing strip. A suitable landing strip is chosen on the basis of selected land cover values, terrain roughness, and terrain elevation models.

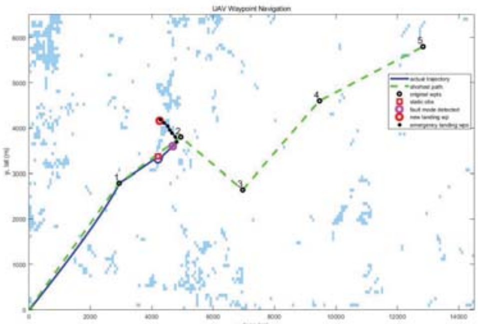

Figure 18. Population density map and UAV trajectory: UAV engine failure mitigation and static obstacle avoidance, using Decision Making BT. An emergency landing strip is found. Obstacle avoidance is achieved using FGA.

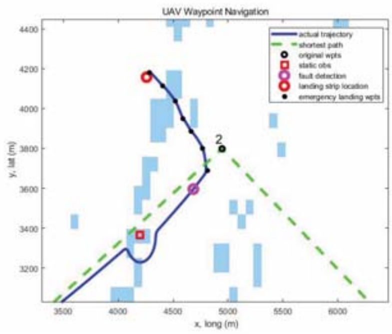

Figure 19. Population density map and UAV trajectory: Emergency landing waypoints generated by LDP.

F. Discussion

We have shown the utility and versatility of the Behavior Tree decision maker in different scenarios consisting of static obstacles such as terrain features, buildings, etc., as well as dynamic obstacles, such as other aircraft or natural fliers. A safe reaction to aircraft engine faults was also shown. In addition, a combined engine failure and stuck rudder were also shown to be mitigated using the BT logic. Hazard mitigation modules were effectively used for mitigation of both faults and obstacles, and the aircraft was able to react in a safe way amid potentially harmful flight situations. Future work for algorithm development includes specifying behaviors for more fault types and additional hazard scenarios. Additional databases may be used for providing more features to the decision logic.

V. Conclusion

A framework for operational safety of UAS has been developed. This formulation is based on Behavior Tree decision making. Detection and Mitigation BTs use a comprehensive and reactive approach to flight anomalies. This framework is advantageous in that aeronautical decision making safe behaviors can be designed, formulated and implemented, as control protocols for safe response to hazards. A case study consisting of four different hazard scenarios, showed that the BT logic can tackle aircraft faults, as well as environmental hazards such as static or moving obstacles. It was shown that the UAV was either able to finish its mission plan or was driven to an emergency landing location, or had to safely terminate its flight.

ACKNOWLEDGMENT

The authors would like to thank Mike Briggs and Andrew Poissant from Millennium Engineering and Integration Company (MEI) for their support and assistance in this work. The satellite, land cover, terrain ruggedness index, digital elevation models, and Landscan data were provided by MEI. We would also like to thank Ed Carney for support with the LDP module, Edward Mortimer for contributions with the FGA simulink module, and Zijie Lin for work with FGA; at University of Maryland, College Park.

REFERENCES

[1] K. Máthé and L. Buşoniu, "Vision and control for uavs: A survey of general methods and of inexpensive platforms for infrastructure inspection," *Sensors*, vol. 15, no. 7, pp. 14 887–14 916, 2015.

[2] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the sky: Leveraging uavs for disaster management," *IEEE Pervasive Computing*, vol. 16, no. 1, pp. 24–32, 2017.

[3] A. A. Doshi, A. J. Postula, A. Fletcher, and S. P. Singh, "Development of micro-uav with integrated motion planning for open-cut mining surveillance," *Microprocessors and Microsystems*, vol. 39, no. 8, pp. 829–835, 2015.

[4] A. M. Samad, N. Kamarulzaman, M. A. Hamdani, T. A. Mastor, and K. A. Hashim, "The potential of unmanned aerial vehicle (uav) for civilian and mapping application," in *2013 IEEE 3rd International Conference on System Engineering and Technology*. IEEE, 2013, pp. 313–318.

[5] M. Huerta, "Integration of civil unmanned aircraft systems (uas) in the national airspace system (nas) roadmap," *Federal Aviation Administration, Retrieved Dec*, vol. 19, p. 2013, 2013.

[6] S. D. Intelligence, "The global uav payload market 2012-2022," *Strategic Defence Intelligence: White Papers*, 2013.

[7] J. S. Duncan, "Advisory circular 107-2: Guidance for conducting suas operations in the nas in accordance with title 14 of the cfr part 107," *US Department of Transportation, FAA*, pp. AFS–800, 06–21–2016.

[8] T. Hutchings, S. Jeffryes, and S. Farmer, "Architecting uav sense & avoid systems," in *2007 Institution of Engineering and Technology Conference on Autonomous Systems*. IET, 2007, pp. 1–8.

[9] I. A. Wilson, "Integration of uas in existing air traffic management systems connotations and consequences," in *2018 Integrated Communications, Navigation, Surveillance Conference (ICNS)*. IEEE, 2018, pp. 2G3–1.

[10] J. S. Duncan, "Advisory circular 90-48d: Pilots' role in collision avoidance," *US Department of Transportation, FAA*, pp. AFS–800, 04–19–2016.

[11] K. W. Williams, "A summary of unmanned aircraft accident/incident data: Human factors implications," Federal Aviation Administration Oklahoma City OK Civil Aeromedical Inst., Tech. Rep., 2004.

[12] S. A. Shappell and D. A. Wiegmann, "The human factors analysis and classification system–hfacs," 2000.

[13] D. Pam, "385-40 army accident investigation and reporting," *Cited in paras*, pp. 2–1, 1994.

[14] T. C. Accardi, "Advisory circular 60-22: Aeronautical decision making," *US Department of Transportation, FAA*, pp. AFS–820, 12–13–1991.

[15] K. W. Williams, "An investigation of sensory information, levels of automation, and piloting experience on unmanned aircraft pilot performance," Federal Aviation Administration Oklahoma City OK Civil Aeromedical Inst., Tech. Rep., 2012.

[16] S. Jacobson, "Aircraft loss of control causal factors and mitigation challenges," in *AIAA Guidance, navigation, and control conference*, 2010, p. 8007.

[17] M. Silambarasan, E. Balasubramanian, and G. Surendar, "Adaptive fault tolerance flight controller for aircraft actuator failure," in *Innovative Design, Analysis and Development Practices in Aerospace and Automotive Engineering (I-DAD 2018)*. Springer, 2019, pp. 527–535.

[18] Z. Lin, L. Castano, and H. Xu, "A fast obstacle collision avoidance algorithm for fixed wing uas," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2018, pp. 559–568.

[19] E. Carney, L. Castano, and H. Xu, "Determination of safe landing zones for an autonomous uas using elevation and population density data." in *AIAA Scitech 2019 Forum*, 2019, p. 1060.

[20] A. Poissant, L. Castano, and H. Xu, "Ground impact and hazard mitigation for safer uav flight response," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2018, pp. 1075–1084.

[21] A. Panta, S. Watkins, and R. Clothier, "Dynamics of a small unmanned aircraft parachute system," *Journal of Aerospace Technology and Management*, vol. 10, 2018.

[22] A. J. Stolzer, C. D. Halford, and J. J. Goglia, *Implementing safety management systems in aviation*. Ashgate Publishing, Ltd., 2011.

[23] W.-C. Li and D. Harris, "Aeronautical decision making: Instructor-pilot evaluation of five mnemonic methods," *Aviation, space, and environmental medicine*, vol. 76, no. 12, pp. 1156–1161, 2005.

[24] R. Adams and A. Ericsson, "Introduction to cognitive processes of expert pilots," *Journal of Human Performance in Extreme Environments*, vol. 5, no. 1, p. 5, 2000.

[25] N. Johnston and N. McDonald, *Aviation psychology in practice*. Routledge, 2017.

[26] G. A. Klein, *A recognition-primed decision (RPD) model of rapid decision making*. Ablex Publishing Corporation New York, 1993.

[27] R. Loh, Y. Bian, and T. Roe, "Uavs in civil airspace: safety requirements," *IEEE A&E Systems Magazine*, vol. 24, no. 1, pp. 5–17, 2009.

[28] J. Guiochet, M. Machin, and H. Waeselynck, "Safety-critical advanced robots: A survey," *Robotics and Autonomous Systems*, vol. 94, pp. 43–52, 2017.

[29] T. Huntsberger, H. Aghazarian, E. Baumgartner, and P. S. Schenker, "Behavior-based control systems for planetary autonomous robot outposts," in *2000 IEEE Aerospace Conference. Proceedings (Cat. No. 00TH8484)*, vol. 7. IEEE, 2000, pp. 679–686.

[30] S. M. Veres, L. Molnar, N. K. Lincoln, and C. P. Morice, "Autonomous vehicle control systems—a review of decision making," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 225, no. 2, pp. 155–195, 2011.

[31] R. A. Brooks, "Elephants don't play chess," *Robotics and autonomous systems*, vol. 6, no. 1-2, pp. 3–15, 1990.

[32] M. Colledanchise, R. N. Parasuraman, and P. Ogren, "Learning of behavior trees for autonomous agents," *IEEE Transactions on Games*, 2018.

[33] L. Wang, "Computational intelligence in autonomous mobile robotics-a review," in *Proceedings of 2002 International Symposium on Micromechatronics and Human Science*. IEEE, 2002, pp. 227–235.

[34] Z. Cao, L. Cheng, C. Zhou, N. Gu, X. Wang, and M. Tan, "Spiking neural network-based target tracking control for autonomous mobile robots," *Neural Computing and Applications*, vol. 26, no. 8, pp. 1839–1847, 2015.

[35] H. Omrane, M. S. Masmoudi, and M. Masmoudi, "Fuzzy logic based control for autonomous mobile robot navigation," *Computational intelligence and neuroscience*, vol. 2016, 2016.

[36] S. K. Murthy, "Automatic construction of decision trees from data: A multi-disciplinary survey," *Data mining and knowledge discovery*, vol. 2, no. 4, pp. 345–389, 1998.

[37] P. Ogren, "Increasing modularity of uav control systems using computer game behavior trees," in *Aiaa guidance, navigation, and control conference*, 2012, p. 4458.

[38] M. Colledanchise and P. Ögren, "Behavior trees in robotics and ai: an introduction," *arXiv preprint arXiv:1709.00084*, 2017.

[39] P. A. Lindsay, K. Winter, and N. Yatapanage, "Safety assessment using behavior trees and model checking," in *2010 8th IEEE International Conference on Software Engineering and Formal Methods*. IEEE, 2010, pp. 181–190.