ABSTRACT

Title of Thesis: DEVELOPMENT OF A SITUATIONAL AWARENESS AND RESPONSE GUIDANCE MODULE FOR SAFER UAV RESPONSE

Andrew Poissant, Masters of Science, 2018

Thesis directed by: Professor Huan Xu
Institute for Systems Research

As Unmanned Aerial Vehicles (UAVs) become more commonplace, there is a growing need for safer flight control software that allows for the UAV to understand and autonomously react to various unsafe flight conditions. Decision-making software must allow the aircraft to perform tasks such as detect and avoid, as well as detect and respond to critical system failures mid-flight. There is a lack of systems engineering in the development of UAV control software safe enough to allow for integration of UAVs into the National Airspace. This lack of systems engineering is a big reason why UAVs are still too unsafe for everyday use. A model-based systems engineering approach is needed to support system requirements, design, analysis, and verification and validation activities.

In this thesis, we provide a model-based systems engineering approach toward a safety module for UAV control software that will allow for safe UAV integration into the National Airspace. System and simulation requirements and architecture are established, in addition to presentation of a collision detection and avoidance

element, decision engine element, and ground impact hazard mitigation element. Detailed models and algorithms are developed for the ground impact hazard mitigation module. Furthermore, simulation results are presented to show the utility of the ground impact hazard mitigation module, which allows UAVs to react safely in the presence of various critical flight anomalies.

DEVELOPMENT OF A SITUATIONAL AWARENESS AND RESPONSE GUIDANCE MODULE FOR SAFER UAV RESPONSE

by

Andrew Poissant

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Masters of Science
2018

Advisory Committee:
Professor Huan Xu, Advisor
Professor Raymond Adomaitis
Professor Jeffrey Herrmann

Dedication

This thesis is proudly dedicated to my incredible mom and dad. Without their guidance and support I would have never made it to where I am today. I will always be thankful for their endless love and dedication to my success.

Acknowledgments

I would like to express my gratitude and sincerest thanks to everyone who have helped make me the successful researcher, student, and person I am today. It is an honor to graduate from such a prestigious university; something I would not have been able to do on my own.

First I would like to thank my advisor, Professor Huan Xu, for giving me an incredible opportunity to work with such a talented and successful research group. It has been a pleasure working on such a pressing topic in today's society and your guidance was definitely paramount to my success as a graduate researcher.

I would also like to thank Millennium Engineering and Integration, Co. Their support for this project was important in allowing me to make such a significant impact in the field of aviation. I want to specifically thank Mike Briggs, Kerry Wisnosky, Ken Baird, and Adam Pederson for their continuous support of this project.

Great thanks go to my coworkers, Lina Castano and Zijie Lin. I am very thankful that you both took the extra time to explain your work and allowed me to incorporate parts of it in my thesis. A special thanks to Lina for being a terrific source of knowledge and help for me.

Last, but certainly not least, I owe my deepest thanks to my wonderful girlfriend, Melanie. Melanie has been my biggest supporter and fan for years and without her I would have never gained the confidence in myself that I have today. She has always believed in me, which has made me believe in myself. And of course,

a son would be nowhere without his parents. My parents have been terrific role-models my entire life and I am so thankful that I have become a positive role-model to my younger siblings.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ϕ	Roll angle
θ	Pitch angle
ψ	Yaw angle
m	Aircraft mass
g	Acceleration of gravity
D	Drag
L	Lift
T	Thrust
Δh_{turn}	Height loss during the turning phase
$v_{s\phi}$	Sink rate in the turning phase
v_{ϕ}	Airspeed during the turning phase
L_{arc}	Arc length
R	Radius of circle
$d\psi$	Total change in heading
ρ_0	Density of air at sea level
S	Wing surface area
C_{do}	Profile drag
k	Induced drag factor
A_r	Wing aspect ratio
C_L	Coefficient of lift
C_{Lo}	Coefficient of lift at zero angle of attack
α	Angle of attack
$d_{x,t}$	Distance traveled in the x direction during the turning phase
$d_{y,t}$	Distance traveled in the y direction during the turning phase
$d_{x,s}$	Distance traveled in the x direction during the straight level phase
$d_{y,s}$	Distance traveled in the y direction during the straight level phase
D_{glide}	Total ground distance traveled during the straight level phase
h_i	Initial height
CE	Casualty expectation
PF	Probability of failure
PD	Population density
PK	Probability of fatality
s	Shelter factor
AL	Lethal area
L_{air}	Length of aircraft
W	Wingspan
DG	Glide distance
DS	Distance to stop
γ	Glide angle

H_p	Height of an average person
d	Lethal distance

BDD	Block Definition Diagram
COA	Course of Action
DOF	Degree of Freedom
FAA	Federal Aviation Administration
FGIF	Feasible Ground Impact Footprint
FWA	Flexible Window Avoidance
GIHM	Ground Impact Hazard Mitigation
IBD	Internal Block Diagram
LCM	Lifecycle Model
MBSE	Model Based Systems Engineering
NAS	National Airspace System
SAA	Sense and Avoid
SARGM	Situational Awareness and Response Guidance Module
SDA	Sense Detect and Avoid
SOI	System of Interest
TBR	To be Reviewed
UAV	Unmanned Aircraft Vehicle
UCD	Use Case Diagram

Chapter 1: Introduction

1.1 Introduction

Unmanned Aerial Vehicles (UAVs) have become more commonplace in todays society and their utility has become vast. UAVs can be used to survey land, carry out military operations, perform rescue missions, or deliver packages [1]. With so many applications, UAVs must be integrated into the National Aispace System (NAS) in order to effectively and safely use UAVs for these purposes. The reason they are not already in the NAS can be deduced by comparing accident rates of UAVs compared to aircraft already present in the NAS. According to Loh et al., UAVs can have accident rates as high as 32 failures per 100,000 flight hours, 32 times higher than accident rates for small general aviation aircraft, and 3,200 times higher than large airliners [2]. Because UAVs have significantly higher rates of failure, they pose a significantly higher risk on the ground and in the air. The high risk that UAVs pose to infrastructure, other aircrafts, and human safety is the predominant issue preventing UAV integration into the NAS.

Reducing a UAV's risk starts with the technology on-board. There is an extensive body of work focused on UAV shortcomings in technology and capabilities, which must be addressed before UAVs may operate in the NAS. A 2012 Congres-

sional Research Service (CRS) Report attributes the high risk of UAVs to mid-air collisions and ground impact resulting from critical flight anomalies [3]. A flight anomaly is any non-normal flight situation which can result in the UAV becoming uncontrollable and a danger to humans on the ground. A flight anomaly is considered critical if the anomaly is unrecoverable. This could include structural failures, propulsion failures, or failures in flight control system [4]. As a result of the report's findings, the CRS recommended that a safety module that allows the UAV to react safely during these critical flight anomalies is included in its flight capabilities. Moreover, the CRS also recommended on-board technology that allows for avoidance of mid-air collisions.

Mid-air collisions include collisions with other aircraft, buildings, or infrastructure. The Federal Aviation Administration (FAA) proposed a set of regulations that will allow for small UAVs (under 55 lbs) to operate in the NAS. One of those requirements are sense and avoid (SAA) abilities for the UAV. This is to ensure the safety of other traffic, population on the ground, and all infrastructure. The SAA function would need to be activated before a collision avoidance maneuver is needed. It also must avoid collisions using gentle and controlled maneuvers [5].

With mid-air collision avoidance and ground impact minimization functions required, changes to UAV control software needs to be the focus of development efforts. UAV integration into the NAS is dependent on the UAV's flight control software having the ability to sense and avoid other air traffic and infrastructure, autonomously return to base, or determine a safe path to crash or land during the presence of critical flight anomalies. A sense and avoid module and ground impact

hazard mitigation module (GIHM) are two core pieces that need to be incorporated in a UAV's flight control software to ensure safe integration of UAVs into the NAS. A lack of high level systems engineering and model based systems engineering (MBSE) practices have been a primary reason that this all-encompassing safer flight control software has not yet been developed. A formal systems engineering approach is needed if UAV flight control software hopes to become safe enough for the NAS.

The goal of this thesis is to take an MBSE approach towards developing the architecture and requirements of a Situational Awareness and Response Guidance Module (SARGM) that will be integrated with nominal flight control software. The safety module will include the following three sub-modules: a collision detection and avoidance sub-module, decision engine sub-module, and ground impact hazard mitigation (GIHM) sub-module. Furthermore, detailed model development of the GIHM submodule will be presented, in addition to integration of the sub-module with nominal flight control software. Fig. 1.2 in Section 1.3 shows the different parts of SARGM and nominal flight control software in the form of a block flow diagram. The diagram is described in more detail in Section 1.3. The nominal flight control software consists of a mission plan, path planning module, 6-DOF aircraft model, and flight controller. The decision engine and collision detection and avoidance sub-modules were developed by fellow University of Maryland researchers, Lina Castano and Zijie Lin. These two sub-modules will be discussed at a high level and integration strategies with the nominal control software will be presented. The GIHM sub-module is presented at a low level, which includes model development and algorithm logic. Simulations will provide a look into how UAVs react to the

different critical flight anomalies. The system presented will take a significant step towards developing control software safe enough that fatality and failure rates of UAVs are on par with or better than aircraft already operating in the NAS.

1.2 Methods and Objectives

A semi-formal MBSE approach was used in the development of a safer flight control software. A V Lifecycle Model (LCM) was used in developing SARGM. Fig. 1.1 shows the V LCM, of which this thesis addresses concept development to system integration. From Fig. 1.1, many of the phases on the right side of the V LCM impact or are influenced by phases on the left side, and vice versa. This illustrates the iterative nature of the system development based on the V LCM. The first step (concept development and stakeholder requirement phases) was establishing a problem that needed a solution and identifying stakeholders. The primary stakeholders are Millennium Engineering and Integration Company (MEI) and the FAA. Secondary stakeholders are potential customers of the flight control software. These could include companies such as Amazon, Google, etc., or government agencies such as the Navy, Airforce, etc.. Following the stakeholder requirement phase, system requirements were established in the system requirements phase.

Concurrently with requirements development, the context level and system level architectural artifacts for SARGM were created in the preliminary design phase. Once the system architecture was established, the bulk of the system design and

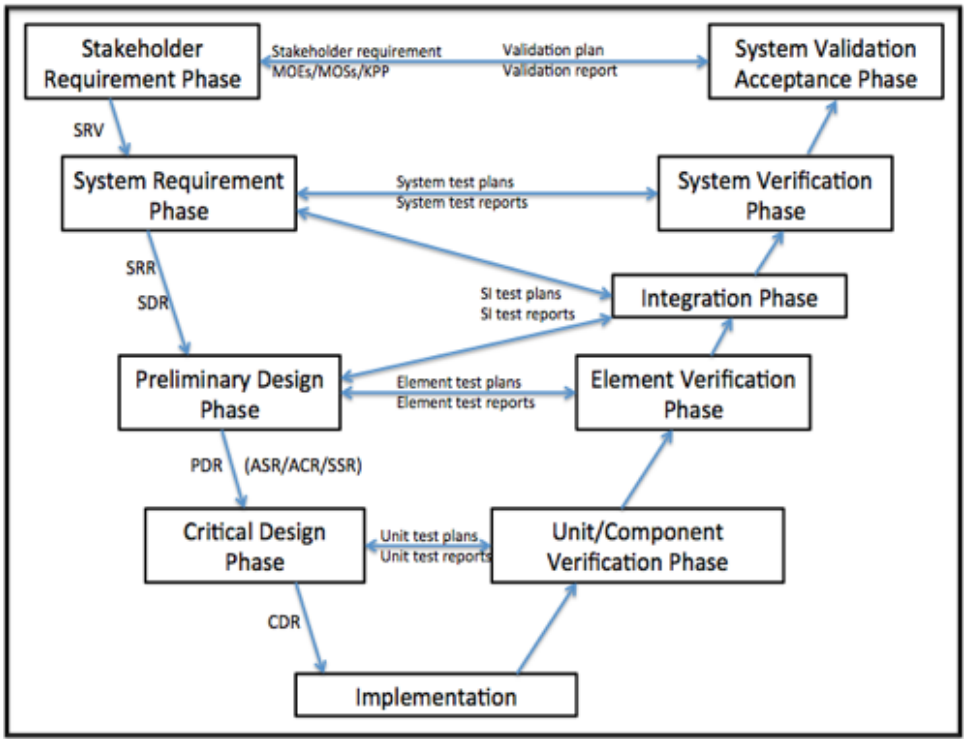


Figure 1.1: V LCM showing the logical project flow for the development of SARGM. This thesis presents work from the Stakeholder Requirements Phase up to the Integration Phase.

development took place in the critical design phase. Following the critical design phase, element verification was performed as part of the element verification phase. SARGM was then integrated with nominal flight control software during the integration phase. System validation, verification, and acceptance was not completed because SARGM is still in the integration phase.

Now that the development approach has been established, it is important to present the objectives. The top-level objectives of SARGM are the following:

1. Reduce rate of failure of any fixed-wing UAV to the level of aircraft already flying in the NAS. Estimate is 2 accidents per one million departures [value

to be reviewed (TBR)] [6].

2. Reduce casualties per flight hour of any fixed-wing UAV to the level of aircraft already flying in the NAS. Estimate is 1 fatality per one million departures [value TBR].

1.3 Concept Description

SARGM contains three elements: a decision engine, collision detection and avoidance module, and ground impact hazard mitigation module. The system will be able to diagnose the following during a mission: if the UAV is experiencing a flight anomaly, what type of flight anomaly is occurring and if it is critical, and whether there is a mid-air collision imminent. SARGM will also have the following functions: determine safest course of action (COA) given a specific flight anomaly or imminent mid-air collision, and update the flight plan to execute the safest COA. The specific situations that SARGM will account for are detailed in Section 2.2.1, where the system requirements are established.

Fig. 1.2 shows a black box representation of how SARGM will be used with nominal flight control software. The nominal flight control software is comprised of the initial set of waypoints, path planning, control system, and aircraft blocks. Everything else in Fig. 1.2 is a part of SARGM. The abbreviation "Dec." represents information flow from the decision engine block. The abbreviations "wps", "coll." and "A/C" mean waypoints, collision, and aircraft/command, respectively. While SARGM includes sensor models and databases, the main functionality of SARGM is

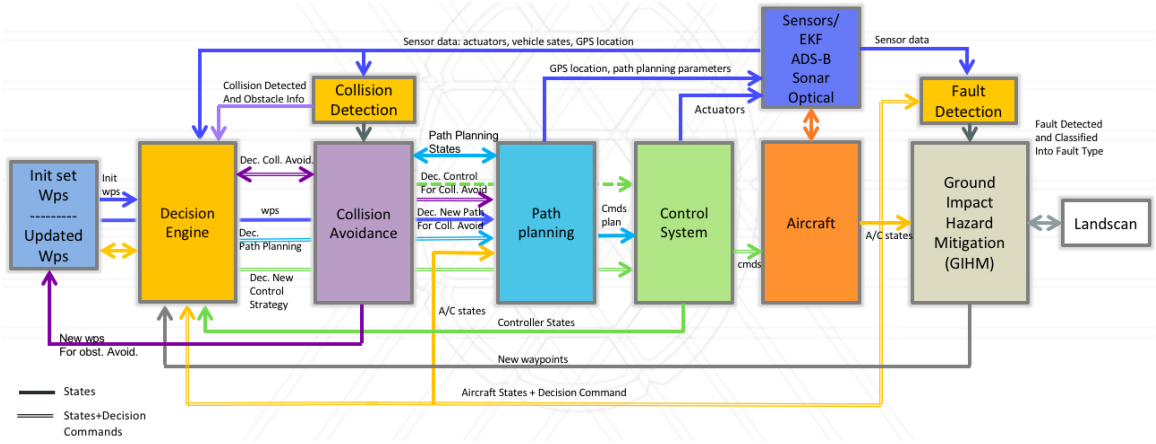


Figure 1.2: Block flow diagram of SARGM integration with nominal UAV flight control software. The initial set of waypoints, path planning, control system, and aircraft blocks constitute the nominal flight control software. Every other block is a part of SARGM.

housed in the decision engine, collision detection and avoidance, and GIHM blocks. Use of the system is designed for the following standard protocol: the user uploads the mission plan in the form of initial mission waypoints and maximum mission flight envelope, the UAV executes its mission, SARGM monitors the UAVs aircraft states and sensor data and determines whether an adjusted flight plan (due to potential mid-air collision or emergency descent resulting from a critical flight anomaly) is needed. If an adjusted flight plan is required, the safest adjusted flight plan is calculated and sent to the flight management system, where the flight management system executes the needed maneuvers. If the maneuver is to terminate flight, the mission ends. If the maneuver is avoiding something in its environment, the flight control software performs the maneuver then returns to its nominal flight plan.

1.4 Background

In developing this system, it is essential to understand the current body work in developing safer UAV control software, in addition to the shortcomings of that body of work. In this section, previous works in safer UAV control architecture and requirements are reviewed. In addition, literature on reachable footprint and ground risk models for UAVs will be presented because reachable footprint and ground risk models are developed in Chapter 3. This section will also discuss the advantages and shortcomings of current work in safer UAV control architecture, ground risk models, and collective risk models.

1.4.1 Safer UAV Control Requirements and Architecture

The development of UAV control software safe enough for integration into the NAS has been a longstanding issue and one that is a challenge because of its complexity. Bogdiukiewicz et al. explains the challenge in achieving this goal arises from the complexity of autonomous decision making mechanisms, and vague and ambiguous safety properties [7]. Loh et al. also noted the need for an integrated requirements collection early in any development program and noted documentation of requirements traceability and rationale is important [8]. This shows that clear and consistent system requirements and architecture is an essential piece of developing safer UAV control software.

To understand what requirements are needed for safer UAV control software, factors which effect UAV safety need to be identified and categorized. Dawei et

al. categorizes these factors as follows: flight platform security (including flying a UAV in a flight safety envelope, engine, body damage, fault, and fault factors), link security (including link interference or loss of communication link), environmental safety (wind, rain, snow, high temperature, cold weather may lead to deterioration of UAV flight performance), and human factors [9]. They also note the flight system platform as the most difficult one to verify, and one that has highest priority. Requirements for safer UAV control software can be encompassed in the three following categories: collision detection and avoidance, decision engine, and ground impact hazard mitigation.

Collision detection and avoidance helps the aircraft avoid any mid-air collisions. The International Civil Aviation Organization explains that the pilot-in-command of a manned aircraft is responsible for detecting and avoiding potential collisions and other hazards, so the same requirement is needed for unmanned aircraft [10]. In addition, right-of-way rules will remain essential in collision detection efforts by unmanned aircraft. Other avoidance requirements noted in [10] include identifying and avoiding terrain, identifying and avoiding severe weather, and providing "visual" separation from other aircraft. A 2012 Congressional Research Service (CRS) Report detailed the required technology and standard procedures for safe UAV control software. The report requires UAVs to have technology and standard procedures for sensing and avoiding other air traffic under all possible scenarios, including loss of communications [3]. In [8], the authors state a requirement driver for any operation in the NAS will be its contribution to maintaining aircraft or UAV separation during flight operations in the NAS. Furthermore, aircraft or UAV safety

for a given phase of flight or surface maneuver is achieved by maintaining minimum separation from other aircraft or UAV and nearby obstacles.

In addition to collision detection and avoidance capabilities, a UAV must have the ability to make on-board decisions during times of unsafe conditions. A decision engine allows the UAV to make mid-flight decisions related to fault detection, fault mitigation course of action, control instability recovery, and risk assessment. Loh et al. claims that to make UAVs safer, it requires substantiation on: (1) the identification of hazards/failure conditions, causes, and effects, (2) assessment of risk, and (3) validation and verification of safety requirements [8]. The fault detection component assesses any potential faults or failures in the overall system. This includes monitoring performance of each sensor, engine operation and structural integrity of the main body and control surfaces, among other things. According to Valavanis, model-based fault detection can be employed to monitor system functionality as well as determining the severity and type of faults or failures [11]. Valavanis goes on to say that if necessary, the fault detection can initiate an emergency maneuver. In addition to fault detection and analysis, [9] cites simulating fault circumstances as part of safety testing for a UAV. If a fault mode is detected by the decision engine and an emergency landing is required, an emergency protocol is needed to determine the safest response. A review of protocol and requirements for this function is provided in the next section, Section 1.4.2.

In addition to clear system requirements, a complete system architecture is needed. [8] and [11] establish a need for clear and concise system architecture when developing safer UAV control software. [8] cites the need for credible architecture

by asserting that credible architecture forms the basis to conduct safety assessments and to demonstrate requirements feasibility.

In developing the architecture, it is necessary to understand the state and limitations of current architecture. Valavanis presents a literature review of challenges in unmanned aircraft autonomy and presents system architecture for unmanned aircraft based on the present challenges [11]. He also explains that there still exists a major drawback when attempting to study, model, develop, and formalize design of systems that operate in uncertain/dynamic environments. This leads to a clear need for a system architecture to help formalize the development of unmanned systems in uncertain environments. Valavanis claims the major challenge of autonomous UAVs has been the level of on-board intelligence and the level of autonomy to be achieved in order to facilitate mission planning, decision making, control execution, data logging, real-time communication to a ground control station, and online mission modification. These capabilities are essential for autonomous operations of an aircraft.

Based on these challenges, [11] established the following basic systems are required of any unmanned aircraft: complete navigation sensor-suite, higher level mission planning and trajectory generation, sensor-based real-time control, failure detection and accommodation, vision capabilities, detect, see/sense-and avoid (DSAA), pilot take-over or assisted flight, and communication capabilities. The on-board sensors include an Inertial Measurement Unit (IMU) and Inertial Navigation System (INS), GPS, a magnetometer for heading information and an altimeter and/or laser range finder for altitude measurements and precision landing capa-

bilities. All of these functions can be comprised in the following three core parts of a safer control software: collision detection and avoidance, decision engine, and ground impact hazard mitigation.

While the works above present different parts of a complete requirements baseline and system architecture, the literature still does not provide a clear and all-encompassing requirements document or complete system architecture. This work presents a clear set of requirements and complete high-level architectural documents for the development of this software.

1.4.2 Ground Impact Hazard Mitigation

A large part of this thesis is focused on the development of models and algorithms for safe UAV response to critical flight anomalies. Because of this, it is necessary to perform a literature review of developing reachable footprint models, generating ground risk profiles, and developing safest response algorithms.

A wide body of literature exists in assessing the collective risk UAVs pose, both in-air [2] [12] [13], and on the ground [12] [13] [14] [15] [16] [17]. Ground risk models can be categorized into the following: failure mode, impact location, recovery, stress, exposure, incident stress, and harm [18]. Of these categories, the most relevant to this work are failure and impact location models.

Characterizing failure modes is an important aspect in guarding against ground impact because if a flight anomaly and the effect of that anomaly on the aircraft states are known, then the aircrafts reachable ground footprint can be calculated

using gliding flight equations. The failure mode determines the capabilities and maneuverability of the aircraft, which must be considered when developing ground footprint models. For UAVs, these modes may include actuator failures, sensor failures, engine malfunction, loss of radio link and GPS, as well as structural damage such as a broken propeller, wing, etc., caused by unanticipated flight events [4]. This work focuses on two particular categories of fault modes: engine malfunction and actuator failures.

Regarding impact location models, previous works have developed reachable ground footprints. [19] investigated the ability of a fixed-wing aircraft to glide to a designated emergency landing area. The authors in [20] and [21] use 6 DOF models to develop ground impact models for determining the reachable ground envelope of a UAVs, and [22] developed an emergency no-thrust flight trajectory plan. None of these works, however, incorporate flight control software into these models. They furthermore do not examine collective risk profiles that consider population data.

To understand the collective risk a UAV poses to a given area, accurate population data can be used. Previous work has used census data or local tax data [12] [13] [14] [15] [16]. However, such information can be hard to acquire and can be largely unrepresentative of the true population count for a given area. [23] used demographic population data for entire cities and states to generate population data for use in developing collective risk profiles. Using data at such large scales can result in great uncertainty when finer resolution population data is required. For this reason, LandScan USA data was used to obtain accurate population data. The LandScan USA dataset represents ambient population at 90 m resolution anywhere

in the United States [24]. In previous literature, Landscan data has been used for mapping global impacts from climate change, building and evaluating population density models, mapping spread of dangerous diseases, and much more [25] [26] [27]. However, to our knowledge, LandScan data has never been used for the purpose of determining the lowest risk ground impact point for a UAV experiencing a hazardous flight anomaly.

While there has been development in collision detection and avoidance, decision making, and ground impact hazard minimization, there is limited work on integrating all three key modules into an all-inclusive flight control software. This work aims at presenting all three modules and using an MBSE approach to integrate all three with nominal UAV flight control software to create a safer and more robust flight control software.

1.5 Outline of Thesis

This thesis takes an MBSE approach towards developing a safer flight control software, with the goal of making UAVs safe enough to be integrated into the NAS. The system of interest (SOI) for this thesis is developed and integrated with an existing nominal flight control software in the upcoming chapters.

Chapter 1 introduces the problem that needs to be solved and establishes how the SOI will help solve the problem. Additionally, it provides background on the body of work already present to help solve the problem. Chapter 2 establishes the system and simulation requirements and architecture for SARGM. It provides a

combination of context level, system level, and simulation architectures to provide a clear picture of the components of the SOI. Chapter 3 develops each of the three elements, or modules, of SARGM. Discussion of the decision engine and collision detection and avoidance will be at a high level because that work was developed by other University of Maryland researchers, and is out of the scope of this thesis. Development of the GIHM module is at a low level and includes discussion of pertinent algorithms, mathematical models, and factors that effect the output metrics.

After module development, Chapter 3 will also discuss how these modules will integrate with each other, and with nominal UAV control software. This includes showing a block diagram of information flows from each part of the all-encompassing flight control software. Chapter 4 will present and discuss the results of software tests. This includes key results from GIHM and end to end simulations of flight missions with the GIHM module integrated with the nominal flight control software. Finally, Chapter 5 will summarize results and provide conclusions and takeaways from the work presented in this thesis. Furthermore, future work will be presented to promote future development of this work.

Chapter 2: Situational Awareness and Response Guidance Module System Design

2.1 Overview

This chapter addresses the system requirements and preliminary design phases of the life-cycle model (LCM). It establishes the high and low level requirements for SARGM. In addition to requirements engineering, key architectural artifacts are presented at the context, system, and simulation levels. These artifacts will be the basis of requirements development and critical design phase execution.

2.2 System and Simulation Architecture

This subsection looks at context level and system level architectures for the system and simulations. Block definition diagrams (BDDs) will show a definition of the system and its environment in terms of its principal elements. Internal block diagrams (IBDs) will show connections between block elements, identify elements and interfaces between blocks established in the BDDs, and display internal structures of blocks. Activity diagrams will show logical data flow behavior, actions to accomplish the stakeholder goals, and how the environment and user interact with

the system. Use case diagrams (UCDs) will show top level functions that stakeholders want and identify external interfaces. Finally, simulation architecture will provide an overview of the structure and behavior of the simulation to aid in future verification and validation activities. The architecture will aid in requirements development, discussed in the next subsection.

2.2.1 Context Level Block Definition Diagram

Context level architecture is the highest level of architecture, which represent all external entities that may interact with the system. Its architecture depicts the system, with no details of its interior structure, surrounded by all its interacting systems, environments and activities. The objective is to focus attention on external factors and events that should be considered in developing requirements and constraints [28].

Fig. 2.1 shows the context level BDD for SARGM. This figure helps us understand the high-level structure of the system's domain, which is comprised of the SARGM system block, user block, and SARGM environment block. Each of these blocks are further decomposed into their components. The SARGM system block is comprised of the collision avoidance, collision detection, ground impact hazard mitigation, and decision engine blocks. The SARGM environment is comprised of the UAV flight management software, sensors, ADS-B data, LandScan data, and UAV flight states blocks.

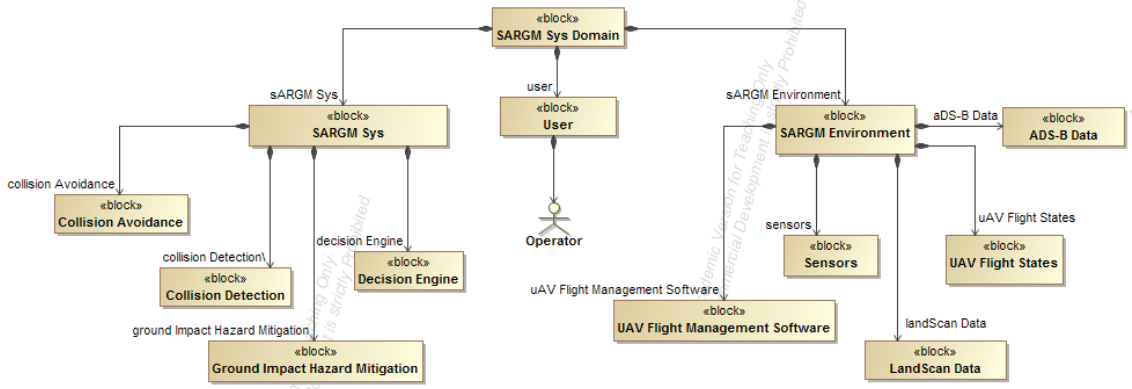


Figure 2.1: Context level BDD for SARGM.

2.2.2 Context Level Internal Block Diagram

Fig. 2.2 shows the context level IBD for SARGM. It helps identify and document interfaces between SARGM and its environment. The diagram also identifies information flows between blocks. For SARGM, the user sends the initial mission plan data to SARGM and receives flight status updates from SARGM. SARGM sends updated mission plan data to the SARGM environment and receives flight data from the environment. The updated mission plan data are new waypoints that must be navigated to. These new waypoints either help the UAV avoid mid-air collisions or is a landing or crashing waypoint for a UAV experiencing a critical flight anomaly. The flight data sent to SARGM is comprised of the updated UAV flight states (velocity, GPS coordinates, altitude, Euler angles, Euler rates, etc.), sensor data, ADS-B data, and LandScan data.

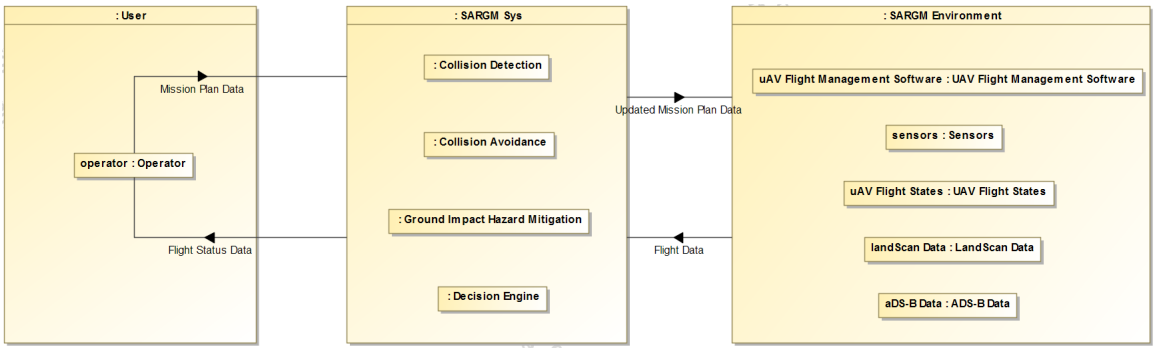


Figure 2.2: Context level IBD for SARGM.

2.2.3 Context Level Use Case Diagram

Fig. 2.3 shows the context level UCD for SARGM. This diagram helps show the scope of SARGM and provides a high-level description. The operator interacts with the decision making element of SARGM by sending initial mission waypoints to it. The LandScan data block provides a high-resolution population dataset to GIHM. ADS-B data and sensors blocks provide data to the collision detection and avoidance module. The UAV flight management software and UAV flight states blocks interface with the decision making module. The decision making module sends mission waypoints to the UAV flight management software and receives flight states data from the UAV flight states block.

2.2.4 Context Level Activity Diagram

Fig. 2.4 shows the activity diagram with swim lanes for the SARGM system. The diagram shows shows the sequences of activities associated with accomplishing

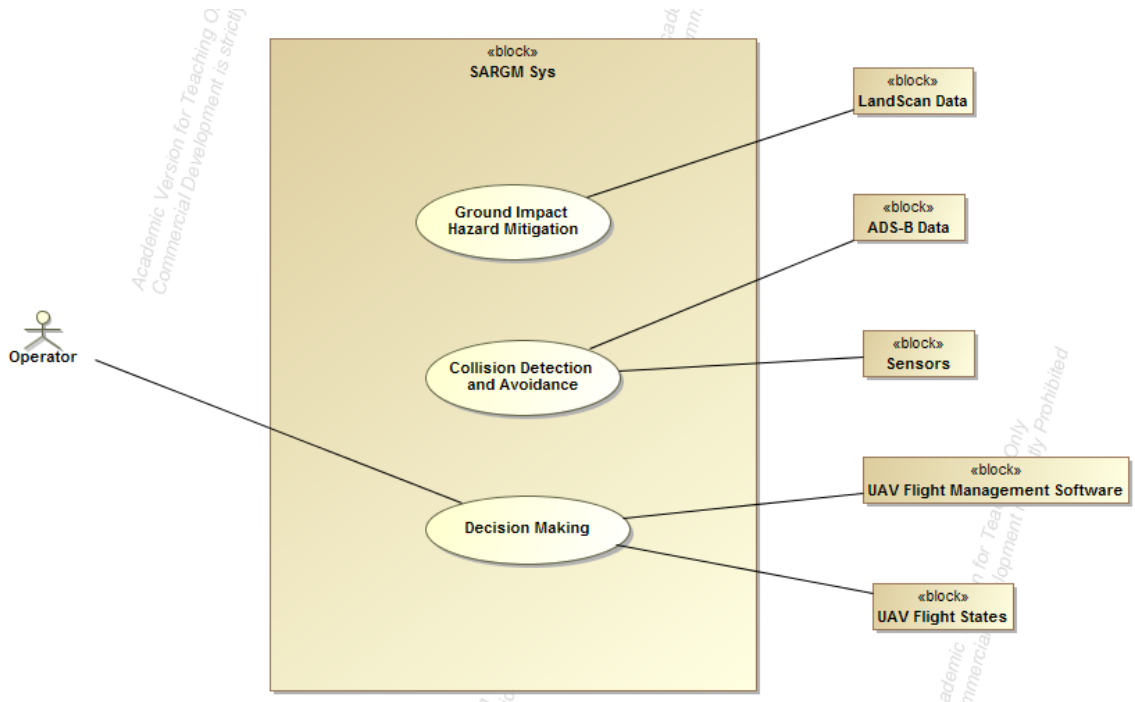


Figure 2.3: Context level UCD for SARGM.

the system's goal of safely operating a UAV. The activity diagram also shows the high level logical flows starting from mission initialization and ending with mission termination. Interface requirements are also depicted through the logical flows between the user, SARGM, and environment swim lanes. The user initiates the process by generating the mission plan and mission flight envelope. The flight management software housed in the environment swim lane executes the mission plan. Finally, SARGM monitors flight performance and generates appropriate courses of action when flight anomalies or potential mid-air collisions are detected.

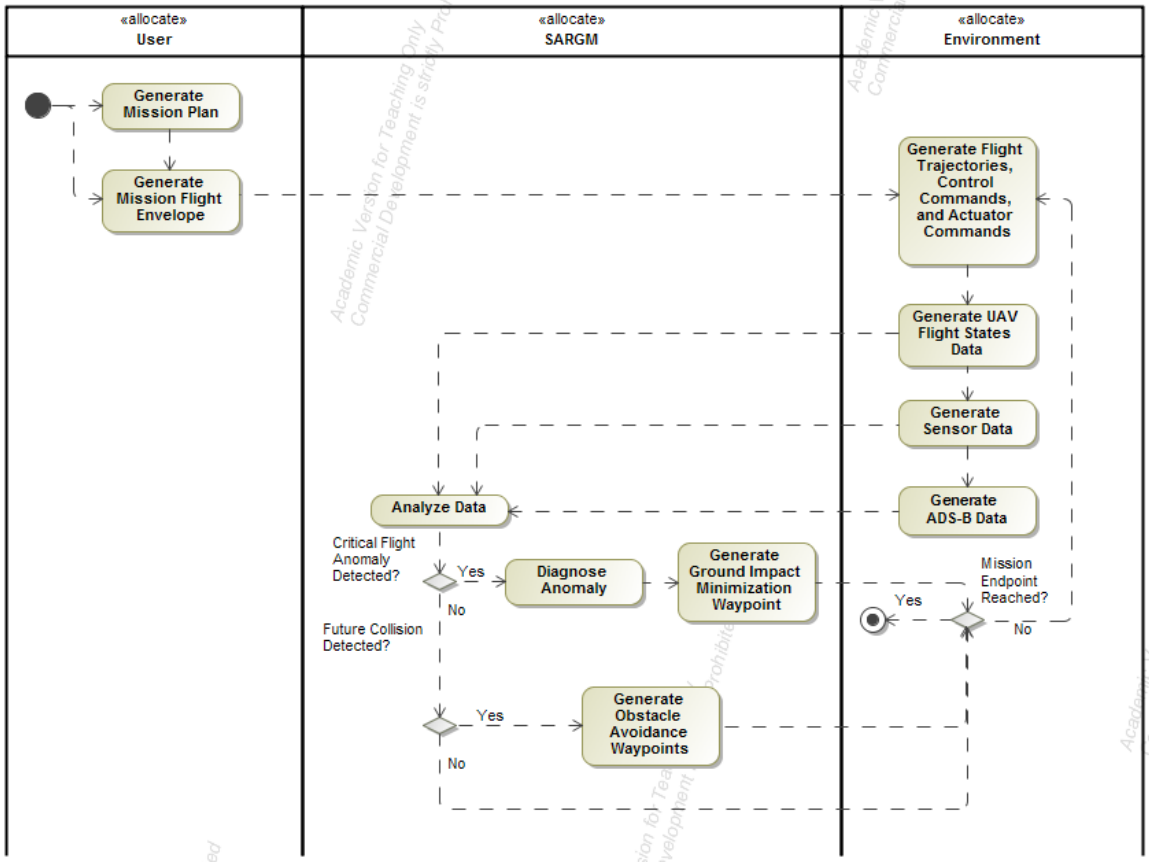


Figure 2.4: Context level activity diagram for SARGM. The black circle represents the starting point and the black circle with a ring around it represents the end point.

2.2.5 System Level Internal Block Diagram

Fig. 2.5 shows the system level IBD. The system level IBD shows how the elements interact with each other to accomplish the system level capabilities. It shows information flows between the SARGM elements, user, and environment. The mission operator provides the mission plan data and recieves mission updates from SARGM. The decision engine provides new priority mission waypoints to the flight management software. These new waypoints are generated when collision

avoidance is required or an immediate landing due to a critical flight anomaly is required. Updated flight states, ADS-B data, sensor data, and LandScan data are provided to the decision engine element.

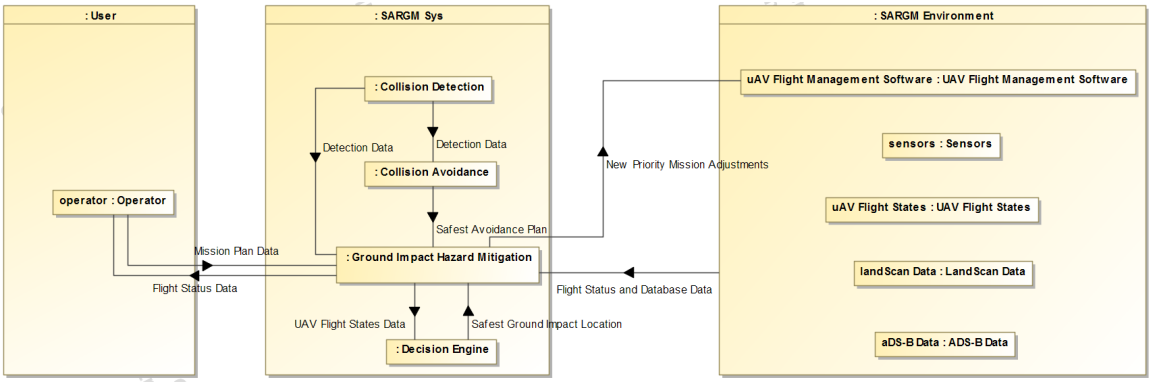


Figure 2.5: System level IBD for SARGM.

2.2.6 Simulation Block Definition Diagram

This subsection provides the architecture for the simulations. This is required for an MBSE process because it provides a framework for developing and evaluating the design and considers the expected behavior of the simulations.

Fig. 2.7 shows the high level structure of the simulations used in verification and validation activities. The diagram indicates the structure of the simulation and its environment. The simulation domain consists of the SARGM simulation, user, and environment. Each of those three blocks are further broken down into their respective parts. The simulations are executed by a systems analyst using MATLAB/Simulink.

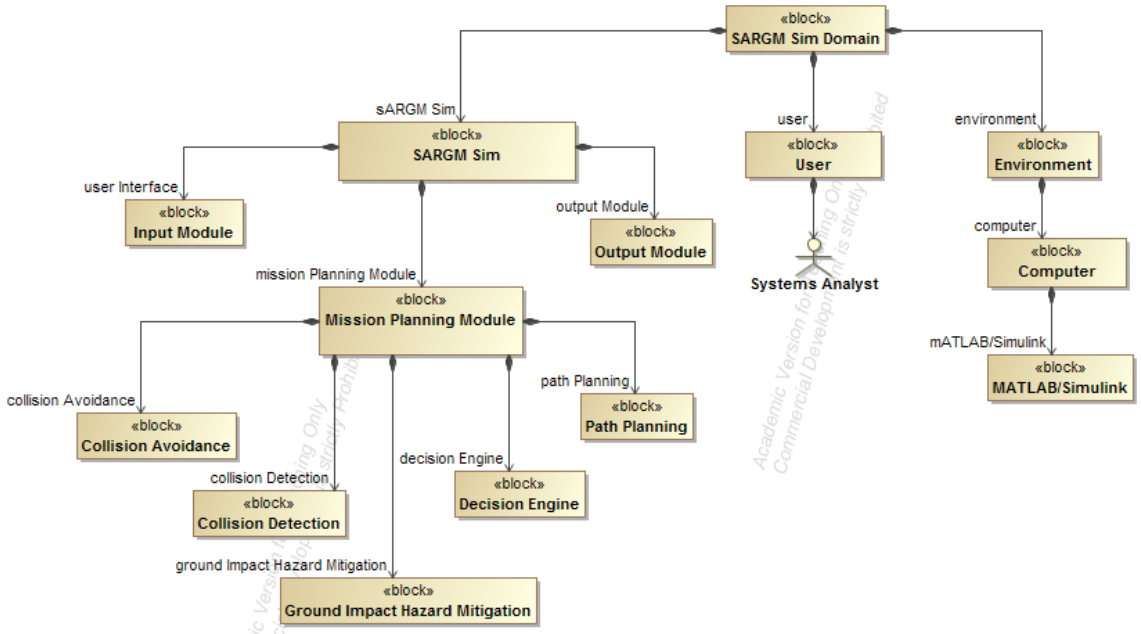


Figure 2.6: Simulation BDD for SARGM simulations.

2.2.7 Simulation Internal Block Diagram

Fig. 2.7 provides the simulation IBD, which shows the system boundary and how the SOI (in this case the simulator) interacts with external systems. Note that SARGM does not directly interact with the systems analyst (user) in the simulations. The user provides input data (detailed in previous diagrams) to the computer. The computer processes the input data and provides simulation input data to the simulation. The simulation runs and provides output data in the form of graphical and numerical outputs. Graphical outputs include trajectory, Euler angles, Euler rates, altitude, and control surface plots. Numerical outputs include run time, decision making time, number and type of flight anomalies experienced, casualty

expectation (fatalities per flight hour), and number of mid-air collisions detected and avoided.

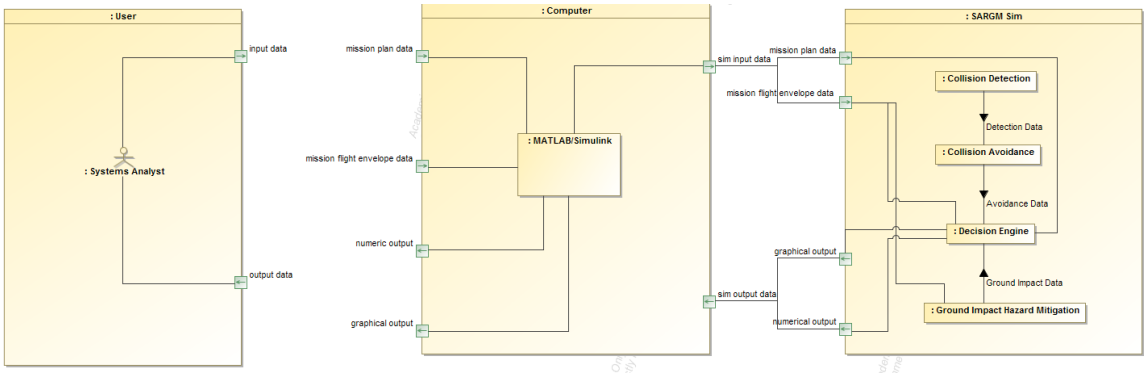


Figure 2.7: Simulation IBD for SARGM simulations.

2.3 System and Simulation Requirements

2.3.1 Requirements

With system and simulation architecture provided, requirements can now be developed. To establish system functionality and interdependence, it is essential to create a requirements baseline. This baseline can serve as a legal document to be referred to throughout the system lifecycle. The requirements for SARGM are established below and are divided into high level, low level, and simulation requirements. High level requirements are for SARGM as a system and low level requirements are for the three individual submodules.

1. SARGM High Level Requirements

    1.1. SARGM shall have a collision detection and avoidance module.

1.2. SARGM shall have a decision engine module.

1.3. SARGM shall have a ground impact hazard mitigation module.

1.4. SARGM shall reduce the rate of failure of a UAV to 1 accident per 100,000 flight hours [value TBR].

1.5. SARGM shall reduce the casualties per 100,000 flight hours of a UAV to 0.1[value TBR].

1.6. SARGM shall integrate with a nominal flight control software containing a mission plan, a path planning module, 6-DOF aircraft model, and flight controller.

1.7. SARGM shall detect any imminent mid-air collision within 300 m of the UAV[value TBR].

1.8. SARGM shall update the mission plan with a new safest response maneuver when a mid-air collision is detected.

1.9. SARGM shall diagnose UAV flight anomalies in real time during mission execution.

1.10. SARGM shall update the mission plan with a new safest response maneuver when a flight anomaly is detected.

1.11. SARGM shall make any mid-flight decision in less than 0.1 second [value TBR].

2. SARGM Low Level Requirements

2.1. Collision Detection and Avoidance Requirements

2.1.1. The collision detection and avoidance module shall detect static or dynamic obstacles within 300 m of the UAV [value TBD].

2.1.1.1. The collision detection and avoidance module shall interface with ADS-B data to obtain real time flight data of all aircraft in its flight envelope.

2.1.2. The collision detection and avoidance module shall avoid at least 4 dynamic non-maneuvering threats whose velocities and trajectories are known.

2.1.3. The collision detection and avoidance module shall avoid at least 4 static non-maneuvering threats.

2.1.4. The collision detection and avoidance module shall provide a safe path that avoids detected threats.

2.1.5. The collision detection and avoidance module shall provide a path to return to nominal mission trajectory after a threat is avoided.

2.1.6. The collision detection and avoidance module shall interface with the 6-DOF model of the nominal flight control software.

2.1.6.1. The collision detection and avoidance module shall obtain UAV flight states (e.g., GPS coordinates, velocity, roll, pitch, yaw) from the 6-DOF model.

2.1.7. The collision detection and avoidance module shall interface with the decision engine by providing an updated mission plan in the form of new waypoints.

2.2. Decision Engine Requirements

2.2.1. The decision engine shall estimate the UAVs augmented state vector.

2.2.2. The decision engine shall diagnose occurrence of flight anomalies.

2.2.2.1. The decision engine shall identify occurrence of a stall.

2.2.2.2. The decision engine shall identify occurrence of ailerons control surface failures.

2.2.2.3. The decision engine shall identify occurrence of rudder control surface failure.

2.2.2.4. The decision engine shall identify occurrence of elevator control surface failure.

2.2.2.5. The decision engine shall identify occurrence of engine failure.

2.2.2.6. The decision engine shall identify occurrence of communication loss.

2.2.3. The decision engine shall monitor engine performance.

2.2.3.1. The decision engine shall calculate total flight time left based on engine performance.

2.2.4. The decision engine shall obtain UAV flight states (e.g., GPS coordinates, velocity, roll, pitch, yaw) from the 6-DOF model of the nominal flight control software.

2.2.5. The decision engine shall identify root cause of flight anomalies.

2.2.6. The decision engine shall determine whether a detected flight anomaly is critical.

2.2.7. The decision engine shall interface with the ground impact hazard mitigation module if an uncontrollable flight anomaly is detected.

2.2.8. The decision engine shall interface with the collision detection and avoidance module if a mid-air collision is detected.

2.3. Ground Impact Hazard Mitigation Requirements

2.3.1. The ground impact hazard mitigation module shall interface with the decision engine to determine which fault mode the UAV is experiencing.

2.3.2. The ground impact hazard mitigation module shall obtain UAV flight states (e.g., GPS coordinates, velocity, roll, pitch, yaw) from the 6-DOF model of the nominal flight control software.

2.3.3. The ground impact hazard mitigation module shall predict the feasible ground impact footprint (FGIF) real-time during mission execution.

2.3.4. The ground impact hazard mitigation module shall use the FGIF to extract local population count map from LandScan dataset.

2.3.5. The ground impact hazard mitigation module shall process LandScan local population count map to extract lowest hazard zones as a candidate for crashing or landing.

2.3.6. The ground impact hazard mitigation module shall select the lowest hazard response.

2.3.7. The ground impact hazard mitigation module shall generate a revised

UAV flight plan to implement safest response when a critical flight anomaly is detected.

3. SARGM Simulation Requirements

3.1. Simulation Capability Requirements

3.1.1. SARGM simulation shall calculate the casualty expectation when requiring an emergency landing.

3.1.2. SARGM simulation shall calculate percentage of lives saved, comparing where the aircraft would have landed with and without SARGM, when requiring an emergency landing.

3.1.3. SARGM simulation shall navigate to the initial waypoints provided by the user, unless a mid-air collision or flight anomaly is detected.

3.1.4. SARGM simulation shall provide a trajectory plot at the end of the simulation.

3.1.4.1. SARGM simulation shall show the user the new landing waypoint on the trajectory plot when requiring an emergency landing.

3.1.5. SARGM simulation shall provide an altitude plot at the end of simulation.

3.1.6. SARGM simulation shall provide a Euler angles plot at the end of simulation.

3.1.7. SARGM simulation shall provide Euler rates plot at the end of simulation.

3.1.8. SARGM simulation shall provide control surface plots at the end of simulation.

3.1.9. SARGM simulation shall tell the user if the aircraft experienced any flight anomalies.

3.1.10. SARGM simulation shall tell the user if the aircraft had to avoid any aircraft or infrastructure.

3.2. Simulation External Interface Requirements

3.2.1. SARGM simulation shall interface with the user.

3.2.1.1. SARGM simulation shall process parsed LandScan data, specifying the mission flight envelope, from the user.

3.2.1.2. SARGM simulation shall take in specific time and fault mode type from the user before simulation execution.

3.2.1.3. SARGM simulation shall process number and location of dynamic and static obstacles from the user before simulation execution.

3.2.1.4. SARGM simulation shall process total simulation time from the user before simulation execution.

2.3.2 Measures of Effectiveness

With requirements developed, it is important to define key performance metrics that indicate success or failure of the system in meeting the stakeholder's needs. Measures of Effectiveness (MOEs) are crucial for any system because these are in-

strumental in determining whether the system will meet the stakeholders' needs. More specifically, MOEs are measures designed to correspond to accomplishment of mission objectives and achievement of desired results. They quantify the results obtained by a system and may be expressed as probabilities that the system will perform as required [29]. The SOI is evaluated by its successful completion of the following MOEs:

- The system can monitor aircraft flight states.
- The system can determine whether the aircraft is experiencing a flight anomaly.
- The system can choose the lowest hazard response when experiencing a critical flight anomaly.
- The system can safely avoid buildings and other aircraft during entire mission duration.
- The system can detect dynamic non-maneuvering threats whose trajectories and speeds are predictable by the sensor system.
- The system can detect static threats such as buildings, trees, etc.
- The system can avoid at least four [value TBR] simultaneous collisions that are predicted.
- The system can generate new priority waypoints to avoid mid-air collisions and collisions with humans on the ground.

- The system can safely land the aircraft when experiencing any critical system failure (engine failure, control surface failure, loss of communication link).

Chapter 3: Situational Awareness and Response Guidance Module Development and Integration

3.1 Overview

This chapter gives a top-level look into the collision detection avoidance and decision engine elements of SARGM. Fig. 3.1 shows a block diagram of how the three SARGM elements interface with the nominal flight control software. This diagram also shows input/output data flows for each block in the control software. This diagram is described in more detail in Section 3.5. Data and logical flows are provided for the collision detection and avoidance element and high level functions for the decision engine are established in this chapter. Following, detailed models are developed for GIHM and uses of the models for various critical flight anomalies are discussed. This chapter will end with discussion about how these three elements of SARGM interface with nominal flight control software.

3.2 Collision Detection and Avoidance

This section provides a top level look at the collision detection and avoidance algorithm developed by Zijie Lin for SARGM. Data flow and algorithm flow are

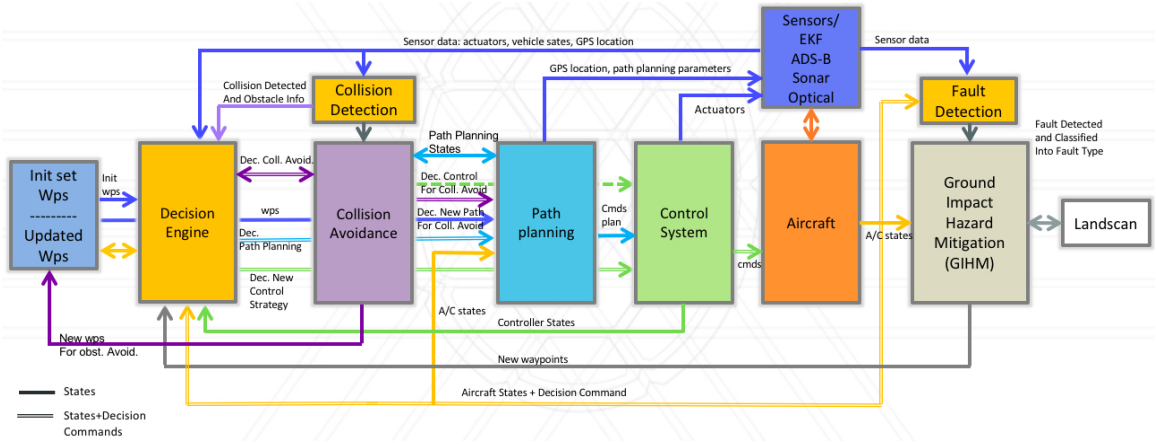


Figure 3.1: Top-level architecture detailing SARGM integration with nominal UAV flight control software, sensors, and databases. The nominal UAV flight control software is comprised of the path planning, control system, and aircraft blocks. SARGM is comprised of all other blocks in the diagram.

presented to provide a clear understanding of the requirements and inner-workings of the algorithm. Then, high level model development and key implications of the work are presented.

3.2.1 Algorithm Data and Logical Flows

Fig. 3.2 provides the response model for the collision detection and avoidance module. The response model gives a black box approach as to what factors (inputs) are required for the model and the resulting metrics (outputs) the model provides. The model requires the following factors: UAV position, UAV flight states, UAV minimum turn radius, sensor data, and obstacle heading. Given this input data, the model computes the following metrics: computation time and avoidance waypoints. The model monitors obstacle location, speed, and heading real-time to calculate

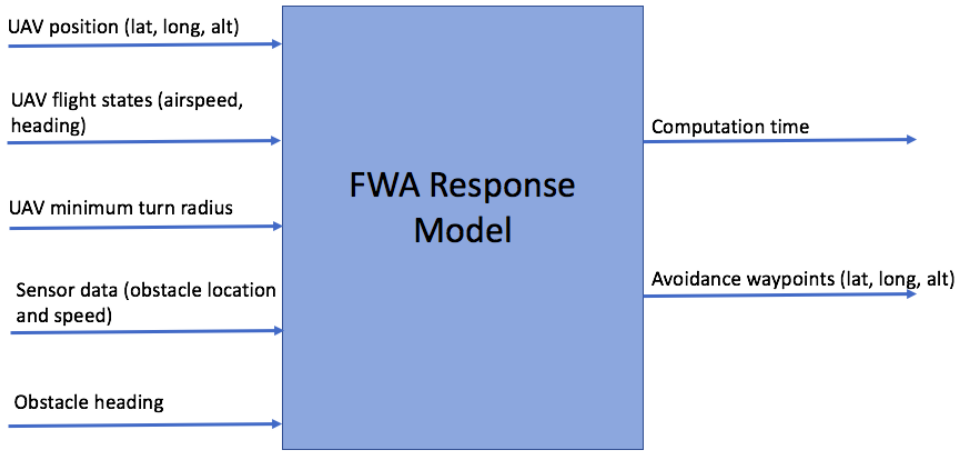


Figure 3.2: Response model for the Flexible Window Avoidance (FWA) collision detection and avoidance algorithm.

whether a mid-air collision is imminent. The avoidance waypoints are used to provide a trajectory for the UAV to avoid mid-air collisions and then guide the UAV back to the original mission path.

Fig. 3.3 shows the collision detection and avoidance algorithm's top level logical flow as an activity diagram with swim lanes. The diagram also shows how the elements of the collision detection and avoidance module interface with each other and the UAV. After the mission is initiated, aircraft states data and sensor data are sent to the collision detection element. The collision detection element analyzes the flight states and sensor data and determines if an imminent collision is detected. If a collision is detected, it determines whether there is a single or multiple obstacles and if each obstacle is static or dynamic. The collision detection and avoidance module allows for detection and avoidance of 4-5 dynamic and static obstacles at once. The collision avoidance element calculates time to collision and classifies each

obstacle detected as urgent or not urgent. This is to prioritize avoidance efforts. The aircraft's required heading to avoid the obstacles is calculated, which is then used to generate waypoints to avoid the obstacles. After those waypoints are generated, the nearest reachable waypoint on the original mission plan is found. This allows for the UAV to continue on its original mission plan after avoiding the obstacles. Finally, the collision avoidance element generates the new mission plan by combining the waypoints to avoid collision and waypoints generated to return to the original path. These waypoints are sent to the UAV's mission planner for execution. The average computation time of the collision detection and avoidance module is 0.0056 seconds.

3.3 Decision Engine

The role of the decision engine in SARGM is to act as the monitoring system that predicts and diagnoses anything that can go wrong with the UAV throughout its mission. This includes fault detection, fault effects prediction, and control-instability and stall recovery. In addition to these functions, the decision engine sends the new mission waypoints, generated by the collision detection and avoidance and GIHM modules, to the UAV's mission planner. This requires interfacing with the collision detection and avoidance module and GIHM module to determine which new mission waypoints it should send to minimize the risk of the UAV.

The decision engine is being developed by Lina Castano at the University of Maryland. Current efforts have focused on fault detection and controlled descent of the UAV in the presence of critical flight anomalies, or fault modes. Faults are

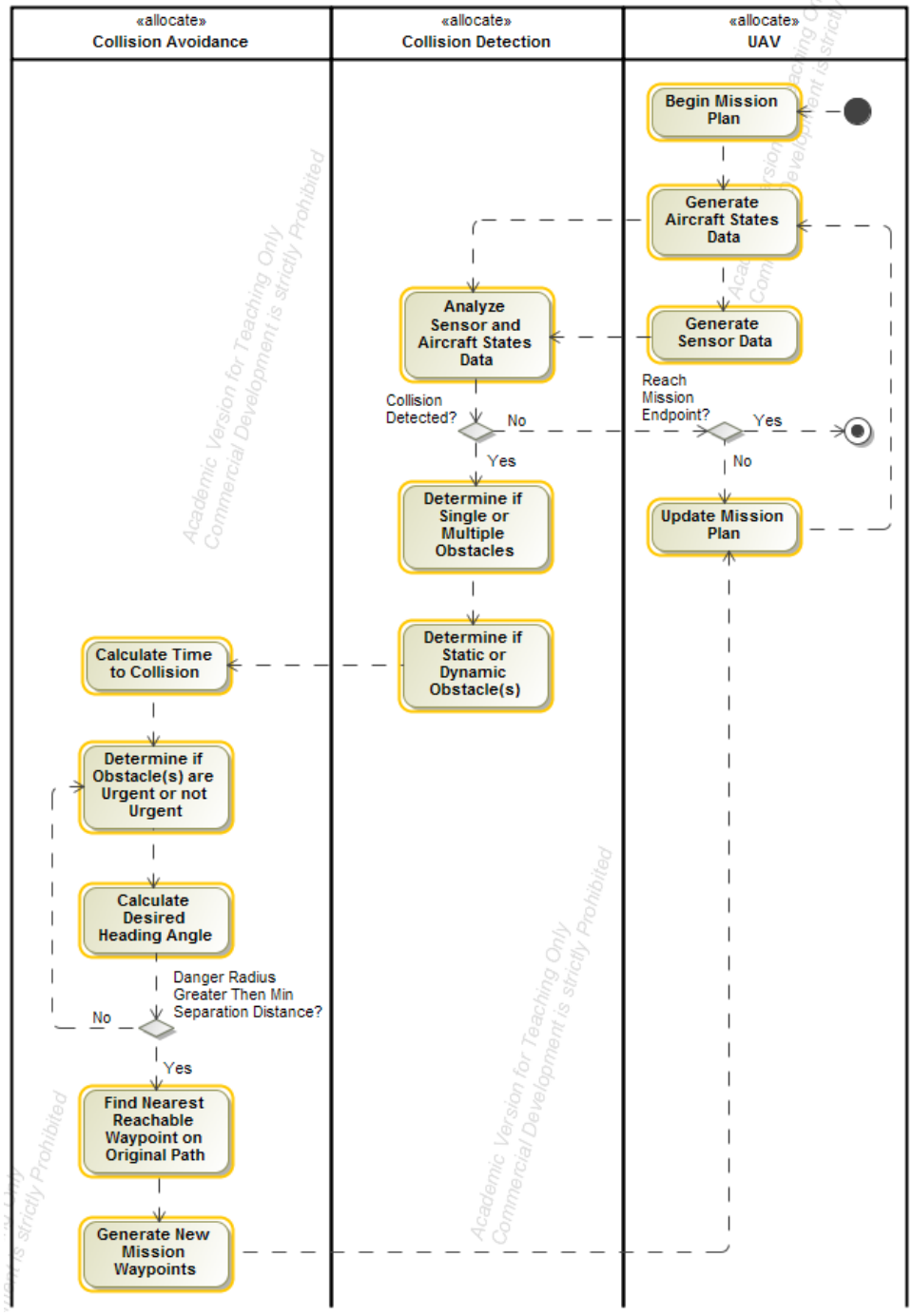


Figure 3.3: Activity diagram for the collision detection and avoidance algorithm. The black circle represents the starting point and the black circle with a ring around it represents the end point.

defined as unpermitted deviations of at least one characteristic property or parameter of the aircraft system from the acceptable or standard condition [30]. Faults are classified according to where they occur in the system (i.e. sensors, actuators and other components). Faults can also be classified as abrupt, incipient, or intermittent, with respect to their time characteristics. The primary fault investigated in this work is abrupt power system failure because of its prevelance in today's UAV failures [31]. In addition, stuck control surfaces (rudder, elevator, and ailerons) were subjects of initial fault investigation.

For fault mode detection, the decision engine monitors actuator commands for the rudder, ailerons, and elevator, and UAV flight states (airspeed, Euler angles, angle of attack, etc.). These factors are used in the determination of whether the UAV is experiencing a fault mode. Once a fault is detected, the fault is characterized as critical or noncritical, where critical flight anomalies are defined as failures that require immediate UAV descent or a flight anomaly that will prevent the UAV from completing its mission safely.

3.4 Ground Impact Hazard Mitigation

Once a critical flight anomaly is diagnosed by the decision engine, the GIHM module determines the lowest hazard response. This section provides a detailed look into the models, collective risk profiles, and decisions required for GIHM. GIHM's primary function is to provide the safest landing or crash point for a UAV in the presence of an critical flight anomaly. First, data flow and algorithm logical flow

are presented to provide a clear understanding of the high level functions and data flows involved with GIHM. Following, the models are derived and fault modes and safest response considerations are introduced and incorporated into the models. GIHM also interacts with the decision engine and collision detection and avoidance elements by the following: the decision engine diagnoses critical fault modes and provides that data to GIHM, and the collision detection and avoidance module interfaces with GIHM by avoiding buildings and other infrastructure during emergency descent. GIHM is unique compared to the current body of work because GIHM integrates specialized reachable footprint models with a high spatial resolution dataset in LandScan to generate accurate ground risk profiles. Furthermore, GIHM algorithms are integrated with flight control software to show how a UAV can maneuver to the new waypoints provided by GIHM for many different critical flight anomalies.

3.4.1 Algorithm Data and Logical Flows

Fig. 3.4 provides the response model for GIHM. GIHM requires UAV position data, UAV flight states data (airspeed, Euler angles, angle of attack), which fault mode the UAV is experiencing (fault modes will be discussed in the next section), and LandScan data. The LandScan data is required to produce collective risk profiles because it provides high resolution population data. GIHM provides the following outputs: casualty expectation (fatalities per flight hour) and new priority mission waypoints. These new priority mission waypoints are waypoints needed to guide the

UAV to the safest landing or crashing zone.

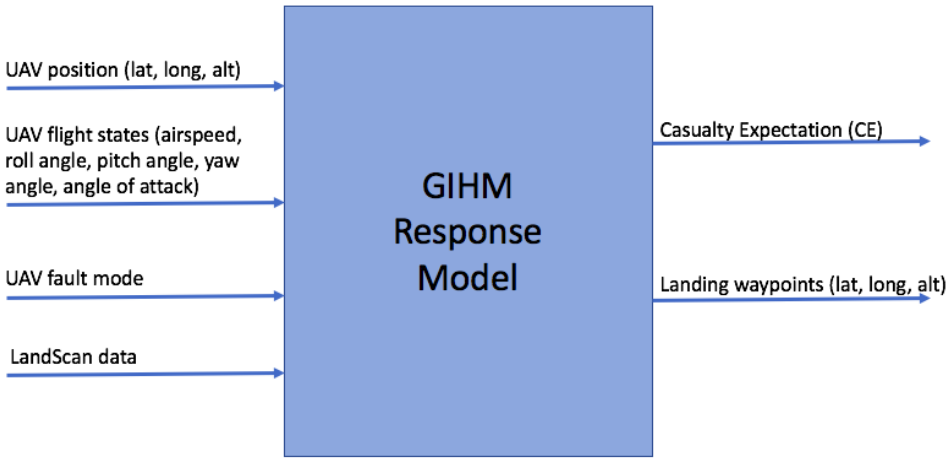


Figure 3.4: Response model for GIHM. Factors are required data into the model and metrics are output data of the model.

Fig. 3.5 shows the algorithm's high level logical flow in the form of an activity diagram with swim lanes. The diagram also shows how GIHM interacts with the decision engine element of SARGM. After the mission is initiated, the decision engine monitors the aircraft states data and sensor data. The data is analyzed for a specified time iteration and the decision engine determines if a critical flight anomaly is detected. If a critical flight anomaly is detected, the decision engine categorizes the flight anomaly and sends that data to GIHM. GIHM chooses which algorithm to use based on which flight anomaly was diagnosed. Depending on the flight anomaly, the level of maneuverability of the UAV may change. From the aircraft's flight capabilities, a feasible ground impact footprint (FGIF) is generated. The FGIF represents everywhere on the ground the UAV can maneuver to. After the FGIF is calculated, LandScan data is integrated with the FGIF to generate

collective risk profiles. These collective risk profiles show the expected fatalities on the ground everywhere within the FGIF. GIHM then determines the lowest ground impact waypoint by searching for the lowest collective risk waypoint. GIHM provides this as the new priority mission waypoint to the mission planner. Finally, the UAV executes the new mission plan by safely maneuvering the aircraft to that waypoint.

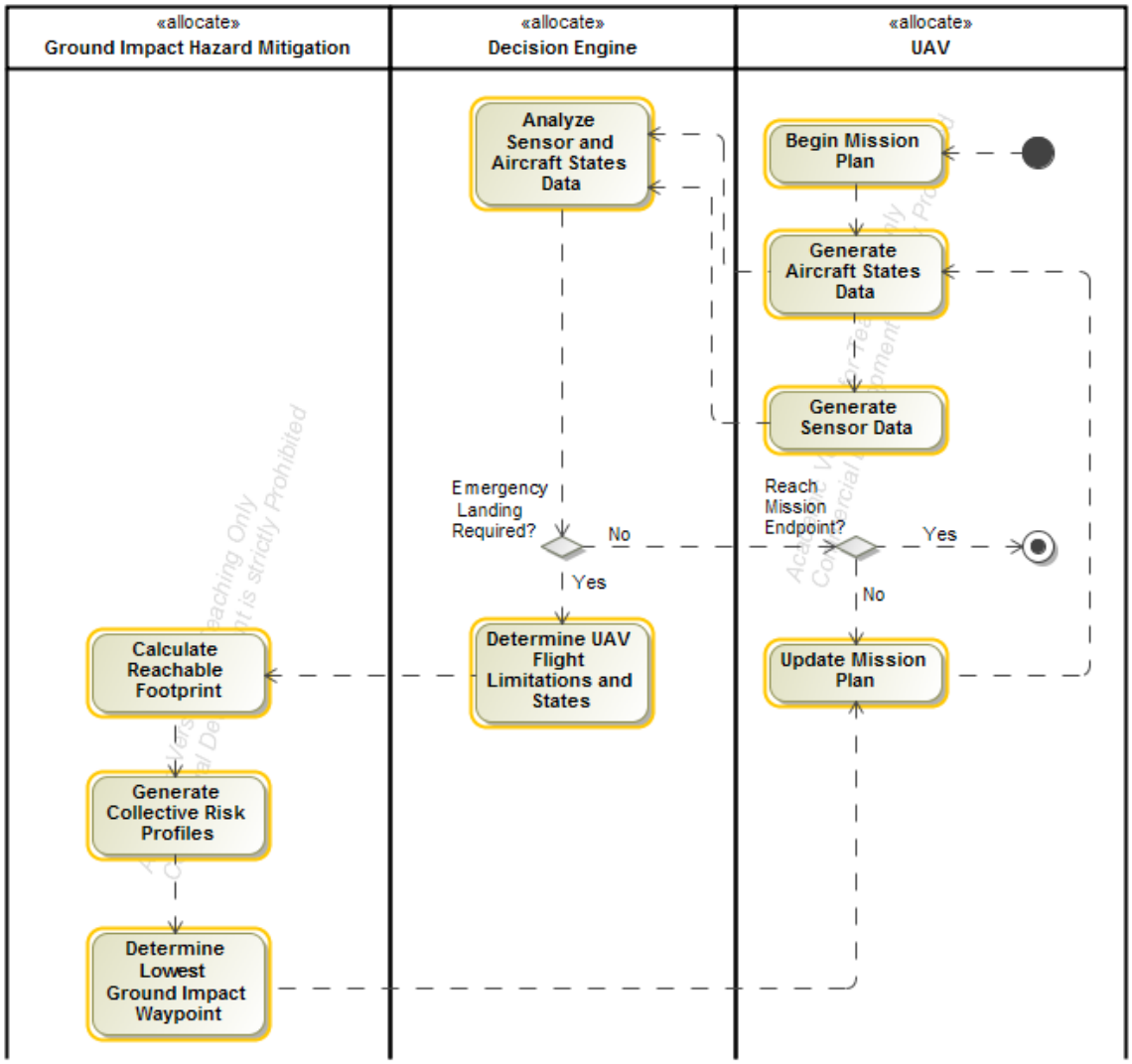


Figure 3.5: Activity diagram for GIHM. The black circle represents the starting point and the black circle with a ring around it represents the end point.

3.4.2 Feasible Ground Impact Footprint Model Development

This section develops the models required to generate an aircraft's FGIF. The FGIF of an aircraft is everywhere on the ground where the UAV can reach. Due to the choice of simulating a prevailing engine failure, glider equations are used for the FGIF calculations. Fig. 3.6 depicts the FGIF concept and axis representation used in development of the FGIF models. Variables d_x and d_y represent displacements in the longitudinal and latitudinal direction, respectively, and d_z represents displacement in altitude. To calculate distance traveled in the x and y directions, gliding flight equations were used to model UAV flight performance. These equations were then iterated over an aircrafts 360^o of maneuverability to obtain a full reachable envelope.

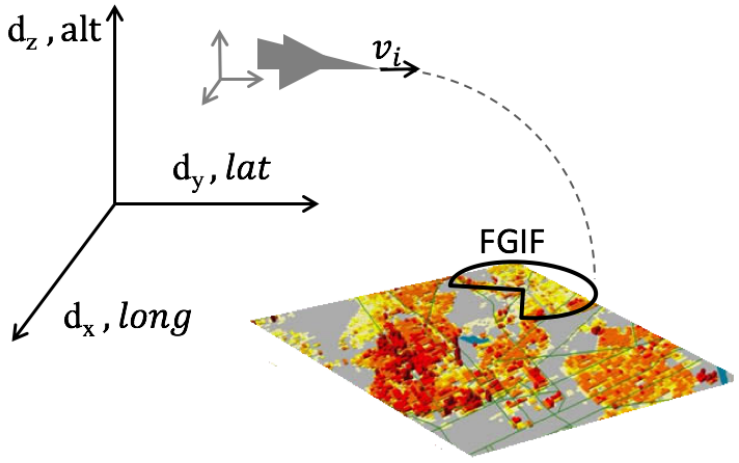


Figure 3.6: Depiction of the FGIF and coordinate system. dx, dy, and dz represent displacement in the longitude, latitude, and altitude directions, respectively. The UAVs reachable footprint is represented by the semi-circle labeled 'FGIF'.

3.4.2.1 Gliding Flight

Calculation of the FGIF requires the vehicles initial latitude, longitude, altitude, airspeed, roll angle, pitch angle, and yaw angle. These values are obtained from the UAV flight control software. The gliding flight equations are derived from the following equations of motion in the aircrafts longitudinal, lateral, and vertical axis respectively [19]:

$$m\frac{dv}{dt} = -mgsin(\theta) - D + Tcos(\theta), \tag{3.1}$$

$$mgcos(\theta)sin(\phi) = mv\dot{\psi}cos(\theta)cos(\phi), \tag{3.2}$$

$$mgcos(\theta)cos(\phi) - L - Tcos(\theta) = -mv\dot{\psi}cos(\theta)sin(\phi), \tag{3.3}$$

where ϕ, θ, and ψ are roll, pitch, and yaw angle, respectively. $\dot{\psi}$ is the turn rate of the aircraft, m is the aircraft's mass, g is acceleration of gravity, D is drag, L is lift, and T is thrust. From here we make the following assumption: for an engine out case, we set $T = 0$, gliding flight. If a small angle approximation is used as a result of a small glide angle (which is the case for most aircraft), the gliding flight equations are simplified to:

$$0 = -mgsin(\theta) - D, \tag{3.4}$$

$$tan(\phi) = \frac{v\dot{\psi}}{g}, \tag{3.5}$$

$$mgcos(\phi) - L = -mv\dot{\psi}sin(\phi). \tag{3.6}$$

These are the three primary equations of motion for gliding flight in the aircraft's longitudinal, lateral, and vertical axis.

3.4.2.2 Footprint Calculation

There are two phases to gliding flight: a turning phase and a straight line phase.

3.4.2.2.1 Turning Phase

During a turn, an aircraft's airspeed will increase and height will decrease. To calculate the height loss during the turning phase, we use

$$\Delta h_{turn} = L_{arc}\frac{v_{s\phi}}{v_{\phi}}, \tag{3.7}$$

where $v_{s\phi}$ is the sink rate of the aircraft in the turning phase, v_{ϕ} is airspeed during the turning phase, and L_{arc} is the arc length of the circle made by the turn. v_{ϕ} and $v_{s\phi}$ are calculated using the following equations:

$$v_{\phi} = vsec^{\frac{1}{2}}(\phi), \tag{3.8}$$

$$v_{s\phi} = v_s sec^{\frac{3}{2}}(\phi). \tag{3.9}$$

L_{arc} is calculated using the following equations:

$$L_{arc} = Rd\psi, \tag{3.10}$$

$$R = \frac{v^2}{gtan(\phi)}, \tag{3.11}$$

where R is the radius of the circle and $d\psi$ is the total change in heading. $d\psi$ is bounded between $\pm\pi$ to account for an aircraft's ability to turn in the positive and negative directions.

An aircraft's rate of sink, v_s, is the amount of height loss per unit time the aircraft is flying during gliding flight. v_s is a function of the aircraft's drag, weight, and velocity through the following equation:

$$v_s = \frac{Dv}{W}. \tag{3.12}$$

To calculate drag, we can modify equations from [32], resulting in the following expression:

$$D = 0.5 C_d \rho_0 v^2 S, \tag{3.13}$$

$$C_d = C_{do} + \frac{k C_L^2}{\pi A_r}, \tag{3.14}$$

where ρ_0 is the density of air at sea level, S is the wing area, C_{do} is the aircraft's profile drag, and k is the induced drag factor, determined by the aircraft wing dimensions, configuration, Reynolds Number and Mach Number. A_r is the aircraft's wing aspect ratio and C_L is the coefficient of lift.

To calculate C_L, it was assumed that the airfoil was cambered and thin. With these assumptions, he following equation was used to calculate C_L:

$$C_L = C_{Lo} + 2\pi\alpha, \tag{3.15}$$

where C_{Lo} is the aircraft's coefficient of lift at zero angle of attack and α is the aircraft's angle of attack. This equation is valid for small angle of attack. Since the aircraft is not approaching stall conditions, the aircraft's angle of attack will be within the linear region of the angle of attack vs. coefficient of lift relationship.

Once the height loss during the turning phase is calculated, we next determine how far in the x and y directions the aircraft travels during the turning phase ($d_{x,t}$,

$d_{y,t}$). Fig. 3.7 shows how $d_{x,t}$ and $d_{y,t}$ can be calculated based on simple geometric ideas. From this figure, it is seen that

$$d_{x,t} = Rsin(d\psi), \tag{3.16}$$

$$d_{y,t} = Rcos(d\psi). \tag{3.17}$$

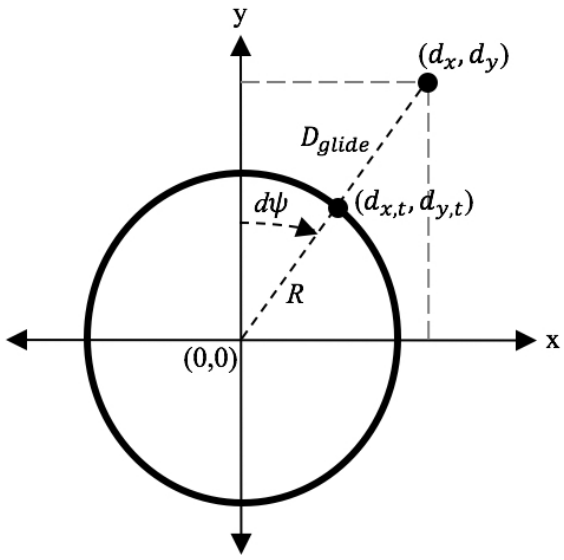


Figure 3.7: Glide range derivation representation, which includes the turning phase and straight line phase.

With the total distance traveled during the turning phase known, the next phase is straight level flight.

3.4.2.2.2 Straight Level Phase

Distance traveled during straight level flight ($d_{x,s}$, $d_{y,s}$) is calculated using similar geometry to that of the turn phase:

$$d_{x,s} = D_{glide} sin(d\psi), \tag{3.18}$$

$$d_{y,s} = D_{glide} cos(d\psi). \tag{3.19}$$

Total ground distance traveled in the straight level flight, D_{glide}, is calculated using the following equation:

$$D_{glide} = (h_i - \Delta h_{turn}) \frac{v}{v_s}, \tag{3.20}$$

where h_i is the aircraft's initial height before entering gliding flight. The loss of height during a turn, Δh_{turn}, must not exceed the aircraft's initial height, which is an important condition when determining where the aircraft can maneuver to.

By adding total distances traveled during the turning and straight line phases, the total distance traveled in each direction can be calculated. This results in the final two equations for distance traveled in the xy directions during gliding flight:

$$d_x = R sin(d\psi) + D_{glide} sin(d\psi), \tag{3.21}$$

$$d_y = R cos(d\psi) + D_{glide} cos(d\psi), \tag{3.22}$$

where d_x and d_y are the coordinates at the end of the glide, relative to the initial position and heading of the aircraft. The FGIF is simply comprised of all of the possible d_x and d_y combinations that the aircraft can reach. It is important to note that for this work it is assumed that there is no wind, which would otherwise have an effect on the gliding performance of a UAV.

Finally, it is important to consider whether an aircraft can actually execute the required turns when finalizing its FGIF. To account for the situation where

the UAV reaches the ground during the turning phase, the following condition is required: if $\Delta h_{turn} > h_i$ then the aircraft cannot execute that turn, resulting in the corresponding d_x and d_y values being excluded in the FGIF.

3.4.2.3 Fault Modes and Safest Response

3.4.2.3.1 Fault Modes

Faults are defined as unpermitted deviations of at least one characteristic property or parameter of the aircraft system from the acceptable or standard condition [30]. Now that models are established for gliding flight, we investigate the effects of four fault modes on gliding performance. The four fault modes investigated are as follows:

1. Engine failure - UAV engine malfunction, resulting in no thrust.
2. Engine and rudder failure - UAV engine and rudder control surface failure. Rudder control surface is stuck at the deflection it had when the fault was detected.
3. Engine and elevator failure - UAV engine and elevator control surface failure. Elevator control surface is stuck at trim deflection value.
4. Engine and ailerons failure - UAV engine and ailerons control surface failure. Ailerons control surfaces are stuck at trim deflection value.

The impact of a fault can be small but it could also lead to overall system failure. After failure detection, a safe autonomous system needs to be able to classify

the fault into an appropriate category in order to mitigate its effects. Faults are classified according to where they occur in the system, i.e. sensors, actuators and other components. Faults can also be classified as abrupt, incipient, or intermittent, with respect to their time characteristics.

In this work, we have considered an abrupt power system fault and a combined abrupt power system/actuator fault. The first actuator fault implemented in simulation consists of a rudder surface that got stuck after servo failure and remain at the deflection it had at the time of the fault. The second actuator fault consists of an elevator surface that got stuck after a servo failure and remains at its trim deflection value. Finally, the third actuator fault implemented in simulation consists of ailerons surfaces that got stuck after a servo failure and remain at trim deflection values [33]. Now we take a look at how the footprint may change depending on which fault occurs.

Fault Mode 1, Engine Failure: In this fault mode, it is assumed that the UAV cannot accelerate but can change its roll, pitch and yaw angles. Because of this, the d_x and d_y equations derived above are used in their entirety to calculate the FGIF.

Fault Mode 2, Engine and Rudder Failure: In this fault mode, it is assumed that the UAV cannot accelerate but can change its heading using the functioning ailerons. Even though the rudder is stuck at a at a specific deflection, it is assumed the UAV may use its ailerons to adjust its heading. It was also assumed the aircraft's sideslip is minimal enough that it could still maneuver to various headings. The validity of this assumption is explored in Section 4.3.2.2, where UAV

trajectory and aircraft states plots are provided for the engine and rudder failure fault mode. Because of these assumptions, d_x and d_y equations derived above are used in their entirety to calculate the FGIF.

Fault Mode 3, Engine and Elevator Failure: In this fault mode, it is assumed that the UAV cannot accelerate or change its angle of attack. The UAV's elevator is stuck at a specific deflection, resulting in the aircraft not having the ability to change its angle of attack. This would result in constant Lift and inability for the UAV to minimize its sink rate by changing its angle of attack. Because of this, d_x and d_y equations derived above are used in their entirety to calculate the FGIF, with the limitation of not being able to change Lift.

Fault Mode 4, Engine and Ailerons Failure: In this fault mode, it is assumed that the UAV cannot accelerate or change heading, but can pitch. Because the UAV cannot change heading, $d\psi = 0$, resulting in only straight line flight because the aircraft cannot execute a turning phase without heading control. From this, the following modified gliding flight equations must be used to reflect that only straight level gliding flight can be achieved:

$$d_x = D_{glide} sin(\frac{\pi}{2} - \psi), \tag{3.23}$$

$$d_y = D_{glide} cos(\frac{\pi}{2} - \psi), \tag{3.24}$$

$$D_{glide} = h_i \frac{v}{v_s}. \tag{3.25}$$

The D_{glide} term was modified to exclude the Δh_{turn} term because the aircraft is unable to turn. Notice that in the equations for d_x and d_y the $d_{x,t}$ and $d_{y,t}$ terms

were removed because the aircraft is not turning while experiencing this fault mode. This results in a straight line reachable footprint.

3.4.2.3.2 Safest Response

With a model developed for the FGIF, we next develop a procedure for choosing the safest response. A searching algorithm was developed, which used the FGIF and UAV's current position to parse the LandScan USA data for the area in which the UAV could maneuver to. It then extracts the local minimum population value for which the UAV would pose the least collective risk. The LandScan USA contains 90 m spatial resolution population data. [N.B. The LandScan USA dataset is restricted for use by government agencies only. We use simulated data at this resolution to show how this dataset can be integrated.] The data structure of LandScan is a matrix whose rows and columns represent latitudinal and longitudinal coordinates. The value of each matrix cell is the population for that range of geodetic coordinates. LandScan data is used to generate a collective risk profile, which is required when determining the safest response.

Collective risk, also known as casualty expectation (CE), describes the aggregate risk that a UAV poses to a population of people. It is measured by expected number of casualties per flight hour [20]. Collective risk is calculated using the following equations [13]:

$$CE = PF \cdot PD \cdot AL \cdot PK \cdot s, \tag{3.26}$$

$$AL = (L_{air} + DG + DS + 2B) \cdot (W + 2B). \tag{3.27}$$

Definitions and the domain of variables for Eqns. 3.26 and 3.27 can be found in Table 3.1. Probability of failure (PF) is expected number of mishaps per flight hour, population density (PD) is population count per square meter, probability of a fatality (PK) is the probability of a piece of the UAV striking a pedestrian and leading to a fatality, and shelter factor (s) is an estimate of how exposed a population is to falling vehicles or debris, with a factor of 0 and 1 representing completely sheltered and completely exposed, respectively. Lethal area (AL) is the area of which a fatality may occur when a vehicle or debris falls. Length (L_{air}) and width (W) refer to the length and wingspan of the aircraft, respectively, buffer (B) is a safety factor, glide distance (DG) is the distance traveled beginning when the UAV is at an altitude of 6 ft and ending when it reaches the ground, and distance to stop (DS) is the total distance from when the UAV reaches the ground to when it comes to a complete stop. Because it is assumed that the UAV will crash and not land, $DS = 0$. Additional research is required to consider landing zones for a UAV.

Lethal area, length, width, glide distance, and distance to stop are all specific to the aircraft and aircraft dynamics. The upper range of population density is determined by the highest population density in the United States, located in Guttenberg, New York City. The resulting casualty expectation values can be as low as 0 fatalities per flight hour and as high as 6.43 fatalities per flight hour. For large airliners, the average casualty expectation is 0.01 fatalities per 100,000 flight hours and that of small general aviation aircraft is 0.1 fatalities per 100,000 flight hours [2]. However, the casualty expectation for a UAV will be much higher than these values because casualty expectation is proportional to probability of failure,

Table 3.1: Definitions of variables in casualty expectation equations. Starred term domain values do not have a range of values because they are specific to the aircraft's dynamics. Representative values were given for the UAV simulated in this work.

Variable	Definition	Domain
CE	casualty expectation	[0, 6.43] fat/flt hr
PF	probability of failure	(0, 1]
PD	population density per square meter	[0, 0.022] pop/m^2
AL	lethal area	77.75 m^2*
PK	probability of fatality	[0, 1]
s	shelter factor	[0, 1]
L_{air}	length	1.83 m*
W	width	1.41 m*
B	buffer	1 ft
DG	glide distance at 6 ft altitude	8.02 m*
DS	distance to stop	0 m

and manned aircraft have very small probabilities of failure, 0.000064%, compared to unmanned aircraft, 2.17% [6] [34]. This leads to a collective risk for small UAVs that is expected to be nearly 100,000 times larger than that of a manned aircraft.

For the purposes of this thesis, probability of failure was assumed to be 0.0217, consistent with the maximum probability of failure for a small UAV defined by Sean et al. [34]. The population density was found by dividing the population count from LandScan data by its respective area. The probability of fatality was found using methods explained by Range Safety Group, which was a function of the mass and speed of the aircraft [13]. A conservative shelter factor of 1 was used, representing a fully exposed population. Glide distance was calculated using the following equations [35]:

$$DG = tan(\frac{H_p}{\gamma}), \tag{3.28}$$

$$\gamma = tan^{-1}(\frac{H_p}{d}), \tag{3.29}$$

where γ is glide angle, H_p is the height of an average person and d is the distance traveled from when the vehicle is at height H_p until when it hits the ground. This glide distance is different than glide distance derived in Section 3.4.2.1 because this is the glide distance starting when the vehicle is at height H_p, not its mission plan height. By knowing the collective risk profile within the FGIF, the UAV is now able to find the point of local minimum risk and decide if the safest response is to fly to that minimum risk waypoint.

While LandScan data is always available, it may not always be required in determining the safest response. If the mission endpoint is within the FGIF of the aircraft then the UAV should naturally land at that waypoint as its the safest response. Similarly, if the mission endpoint is not within the FGIF, but the mission start is, then the aircraft should return to base.

3.5 Integration with Standard UAV Control Software

With all three modules of SARGM presented, we now discuss integration of SARGM with nominal UAV control software. A 6-DOF flight simulation software was implemented in Matlab/Simulink and interfaced with SARGM. Fig. 3.1 shows the block flow diagram for the three elements of SARGM with nominal UAV control software. The nominal flight simulation architecture contains a path planning block, control system block, and aircraft block. The path planning block takes the initial mission waypoints and calculates where the UAV needs to go to reach the waypoints.

It also takes the system states as an input so that it can update where the UAV needs to go. The control system block uses PID controllers to track altitude and groundspeed, and takes the waypoint navigation requirements and outputs actuator commands to the UAV. Finally, the aircraft block houses the 6-DOF UAV aircraft model and takes in actuator commands to update the new system states.

The UAV states, generated from the aircraft block, are sent to the decision engine. The decision engine uses this data and sensor data to determine whether the UAV is experiencing a fault mode. In this work, a fault mode is detected when the UAV aircraft states are outside of a predetermined nominal range. If a fault mode is detected, the decision engine sends this information to GIHM, where it generates new landing waypoints in the form of GPS coordinates. The GPS coordinates are sent back to the decision engine and then to the UAV flight control software. These new coordinates replace old mission waypoints.

Concurrently, the collision detection and avoidance block process the UAV states data, in addition to sensor and ADS-B data, to determine if a mid-air collision is imminent. If a mid-air collision is imminent, the block calculates an avoidance path and a path back to the original mission waypoints. This allows for the UAV to avoid the mid-air collision and then continue in its mission. Those sets of waypoints are sent to the decision engine and then to the UAV flight control software for implementation. The logic behind whether the collision detection and avoidance block or GIHM block takes precedence in sending new priority waypoints was not investigated in this work but will be the subject of future work.

Chapter 4: Simulation Results and Discussion

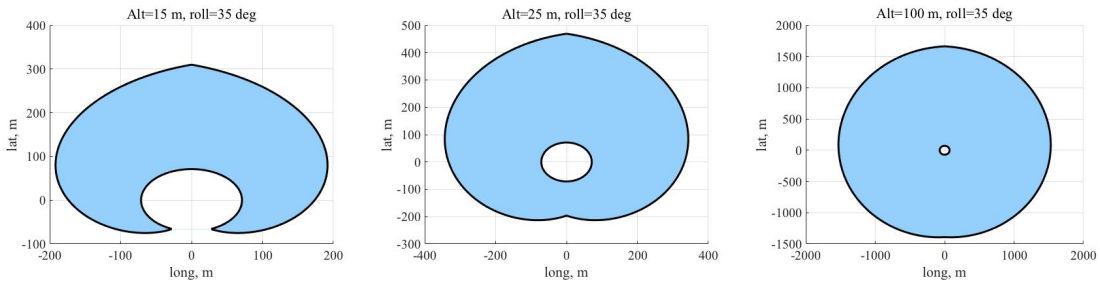
4.1 Overview

This chapter presents simulation results from the GIHM models and end to end simulations of GIHM integrated with nominal control software. The UAV simulated was a Talon 240. The GIHM simulations will establish relationships between some UAV flight states and size and shape of the FGIF. The end to end simulations will provide a look into how UAVs could minimize their ground risk when experiencing critical flight anomalies with the GIHM module. The chapter will finish with discussion and implications of results.

4.2 Ground Impact Hazard Mitigation Results

Section 3.4.2.1 developed the models for generating a UAV's FGIF. This section provides a look into how the FGIF can change with different aircraft states values.

Fig. 4.1 shows the FGIF of an aircraft traveling due north, experiencing fault mode 1 (engine failure) at different heights. The aircraft's initial position when experiencing the fault mode is at the origin of the plots. It can be seen that as the

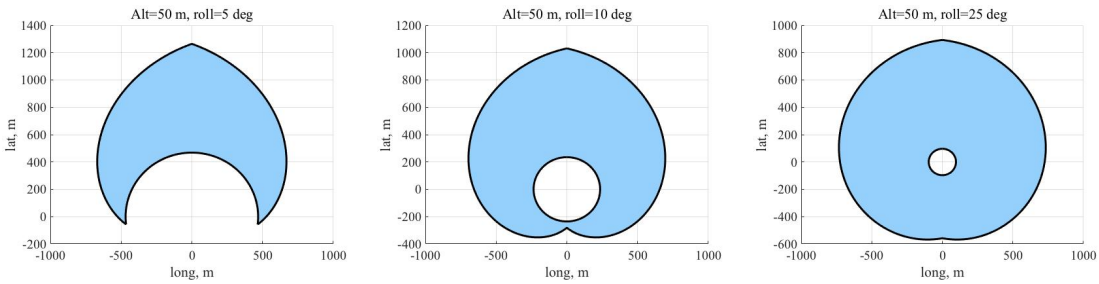


(a) (b) (c)

Figure 4.1: Effect of aircraft height on FGIF for fault mode 1: (a) Alt=15 m, roll=35°. (b) Alt=25 m, roll=35°. (c) Alt=100 m, roll=35°. The light blue shaded area is the FGIF.

initial height of the aircraft increases, so does the FGIF (represented by the light blue shaded area of each plot). This is because the aircraft is able to glide for a longer time, resulting in a larger footprint. Also seen in 4.1, the FGIF's shape changes. As the initial height decreases, the UAV reaches the ground before changing its heading to the full $\pm\pi$ radians of maneuverability.

Fig. 4.2 shows the FGIF of an aircraft traveling due north, experiencing fault mode 1 at different roll angles. At smaller roll angles, the aircraft cannot execute turns to larger heading changes fast enough. As a result, the aircraft cannot reach those larger angles before before reaching the ground. This is seen in Fig. 4.2(a) where the aircraft experiences the fault mode while at a small roll angle. Because of this small roll angle, the FGIF is only part of a circle. As the roll angle increases, the aircraft can more quickly maneuver to larger heading changes. This is seen in Fig. 4.2(b) and Fig. 4.2(c) where the aircraft has a larger roll angle and can reach the entire $\pm\pi$ range of heading change.



(a) (b) (c)

Figure 4.2: Effect of aircraft roll angle on FGIF for fault mode 1: (a) Alt=50 m, roll=5°. (b) Alt=50 m, roll=10°. (c) Alt=50 m, roll=25°. The light blue shaded area is the FGIF.

The FGIF of an aircraft experiencing the engine and rudder failure fault mode would have very similar characteristics as the engine failure fault mode. This is because when the rudder is out, the ailerons are able to change the heading of the aircraft and allow for relatively straight line flight. However, there will be side slip as a result of the rudder control being compromised. For this work, it is assumed that the side slip is minimal enough to where the aircraft can still maneuver to the calculated crash zone.

The FGIF of an aircraft experiencing the engine and elevator failure fault mode would have a similar footprint compared to the engine failure fault mode. However, the FGIF would only be comprised of the black outline (not shaded blue area) of the FGIFs seen in Figs. 4.1 and 4.2. This is because with the elevator out the aircraft is unable to change its angle of attack, so it is unable to change its orientation to reach the shaded blue area. This results in fewer crash zone choices for the aircraft.

The FGIF of an aircraft experiencing the engine and ailerons failure fault mode

would simply be a line. Because there is no turning phase for this fault mode, the UAV cannot change its heading and therefore would only be able to fly at its current heading. However, the aircraft can still change its angle of attack using its elevator, which allows for a straight-line footprint to be made, rather than obtaining a single impact point.

4.3 Integration Testing Results and Discussion

4.3.1 Problem Setup

In this case study, a Talon 240 fixed-wing UAV was modeled. The aerodynamic characteristics used for the small UAV are seen in Table 4.1.

Table 4.1: Characteristics of the General Aviation UAV used in simulation.

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

The UAV's nominal mission plan contains five waypoints and a home waypoint. The waypoints approach the University of Maryland, and a given fault mode is induced 25, 45, 63, 80, 90, and 110 seconds into the simulation. These times were

chosen to obtain a wide range of flight scenarios with the UAV at different points during its mission.

Model LandScan data was created in Matlab and was the same structure as LandScan USA data, but with a resolution of 30 m. The finer resolution was required to show the utility of GIHM for small UAVs because of the size of the FGIF during simulations. Fixed blocks of higher population were created to mimic higher population expected in clusters of buildings at the University of Maryland. The dataset was preloaded into the UAV simulation and was parsed in real time when a fault mode was detected.

4.3.2 Simulation Results and Discussion

4.3.2.1 Casualty Expectation Reduction

This section presents numerical results for the testing of GIHM. Table 4.2 summarizes casualty expectation results of simulations for all four fault modes at six fault mode times. By comparing the average casualty expectation with GIHM to the average casualty expectation without GIHM, we can conclude that the average casualty expectation is 24.718 fatalities per 100,000 flight hours lower with the GIHM module than without the module. This equates to an 97.3% decrease in fatalities per flight hour. These CE values are within the expected range of CE values established in Table 3.1. Note that the units in Table 3.1 for CE is fatalities per flight hour, not fatalities per 100,000 flight hours. According to the FAA, a large airliner shall have a casualty expectation of 1 fatality per 1,000,000 flight hours, which is still

far below the average casualty expectation value for the UAV with GIHM. This is because the probability of failure for this simulation is 2.17%, which is nearly 300,000 times higher than the probability of failure for a large airliner (0.000064%). It is important to note that these results are specific to this mission configuration and are dependent on population differences in the area in which its flying. If a UAV is flying over a rural, low populated area, the percent decrease in fatalities per flight hour as a result of GIHM would be less. From the results, it is concluded that GIHM reduces casualty expectation of a UAV experiencing any of the four fault modes presented in this thesis.

4.3.2.2 UAV Mission Simulations Results

With numerical results presented, it is important to show that the UAV can maneuver to and reach the low hazard waypoint provided by GIHM. This section provides the trajectory and aircraft states profiles for the simulated UAV. The trajectory and aircraft states profiles are presented for when the UAV does not experience any fault modes, and when the UAV experiences fault modes 1, 2, or 4. These results will show the limitations and maneuverability of a UAV experiencing various fault modes. They will also provide evidence that the FGIFs and low hazard waypoint provided by GIHM accurately model where the UAV can glide to.

Fig. 4.3 and Fig. 4.4 show the flight simulation trajectory and altitude profiles of the UAV under nominal operating conditions. The simulation terminates after the UAV reaches the fifth (final) waypoint. Plots of trajectory show the relative

Table 4.2: Simulation results comparing CE with and without the GIHM module for all four fault modes at six different fault times. CE has units of fatalities per 100,000 flight hours.

Fault Mode	Fault Time (s)	CE With GIHM	CE Without GIHM
1	25	0.000	9.690
2	25	0.000	9.690
3	25	0.426	15.250
4	25	1.124	13.940
1	45	0.024	13.350
2	45	0.024	13.350
3	45	0.072	67.450
4	45	2.709	62.560
1	63	0.017	61.650
2	63	0.017	53.900
3	63	0.277	60.280
4	63	1.124	54.130
1	80	0.017	15.760
2	80	0.017	15.760
3	80	0.524	13.239
4	80	3.698	13.780
1	95	0.017	19.400
2	95	0.017	19.400
3	95	1.403	16.960
4	95	4.509	3.928
1	110	0.000	18.140
2	110	0.000	18.140
3	110	0.043	15.860
4	110	0.639	4.336
Average		0.696	25.414

distance the UAV travels compared to the UAV's home waypoint. On all of the trajectory plots, the heat map represents population count in that node, with darker nodes representing highly populated areas. The population heat map is pixilated

because the data was generated using randomized values. The randomized values were generated based on average population values in the University of Maryland area. When LandScan USA data is used, it can easily be integrated and the heat map of the population will be smoother and more accurate. From Fig. 4.3, it is seen that the actual flight path of the UAV is very close to the shortest path, indicating an effective trajectory control scheme. Fig. 4.5 shows the Euler angles and Euler rates profiles for the nominal flight simulation. These will be used for comparison to the flight states of the UAV experiencing the various fault modes.

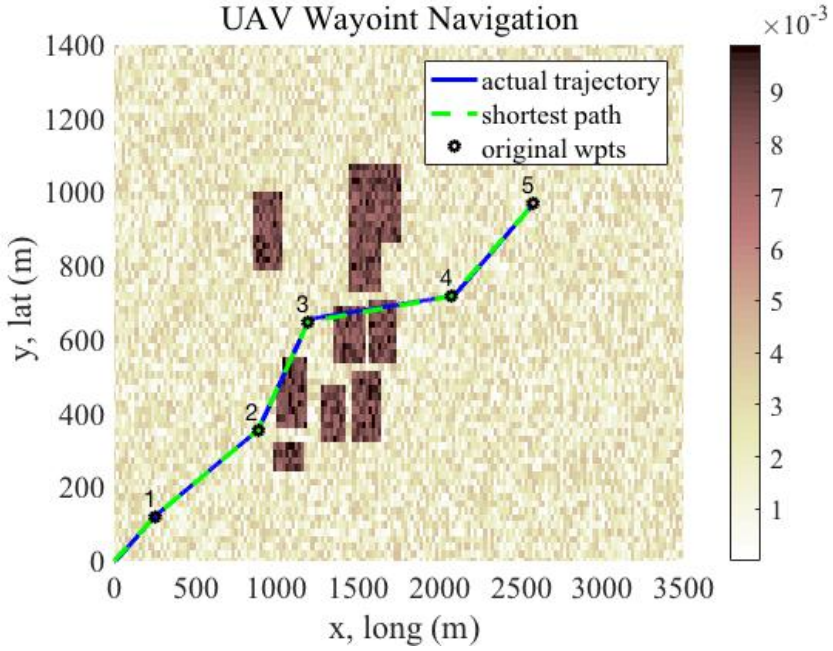


Figure 4.3: UAV trajectory for when there is no fault mode detected. The aircraft starts from the home waypoint at the origin and travels to the waypoints in numerical order.

First, fault mode 1 was investigated. Fig. 4.6 and Fig. 4.7 show the simulated flight trajectory and altitude profiles of the UAV experiencing the engine out (fault

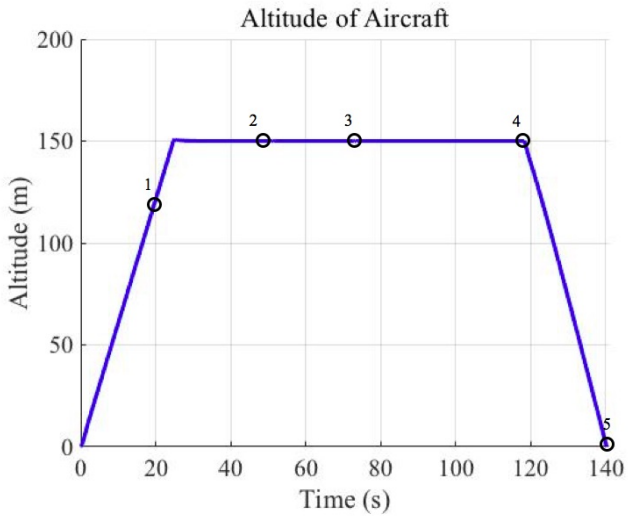


Figure 4.4: UAV altitude plot as a function of time for when there is no fault mode detected. The numbered circles show the five mission waypoints. The UAV begins on the ground at the start, climbs to 150 m, then descends back to the ground when approaching waypoint 5.

mode 1) flight anomaly 80 s after mission plan initiation. The trajectory plot shows where the UAV would have landed compared to where it was able to maneuver. From the heat map, the new landing waypoint is in a less populated spot compared to the old landing point. At 80 s, as the UAV just passes waypoint 3, the engine out fault mode is detected, which results in the change of heading and change in mission plan. Fig. 4.7 also shows the throttle profile of the UAV experiencing fault mode 1 80 s after mission initiation. At 80 s, the throttle is at 0, which is consistent with the condition of the engine out case. From the altitude and trajectory plots, it can be concluded that the UAV found a minimum ground impact point and is flying to that point. A safety factor of 30 m was included so the UAV does not land right next to the highly populated cluster of buildings. Fig. 4.8 shows the

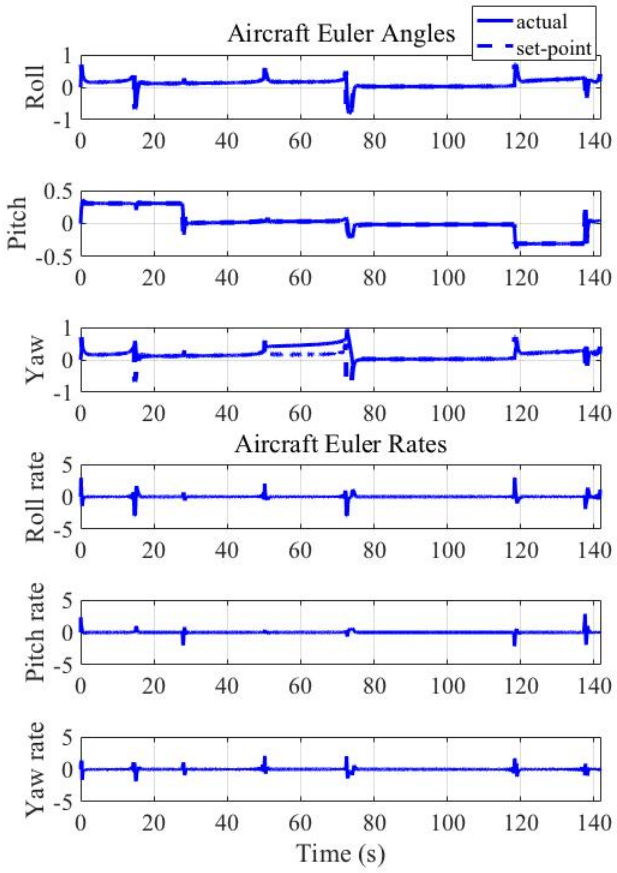


Figure 4.5: UAV Euler angles and rates profiles for when there is no fault mode detected. Angles are in radians. The dashed line on the Euler angle plots show the controller commands and the solid line shows the actual aircraft states.

Euler angles and Euler rates for the UAV experiencing fault mode 1 80 s after flight initiation. Note the significant change in Euler angles at time 80 s. This is a result of the aircraft changing its heading to fly to the new priority waypoint generated by GIHM.

Fig. 4.9 shows the finish mission functionality of GIHM when the end waypoint is within the FGIF. Fig. 4.9 shows the simulated flight trajectory of the UAV

experiencing the engine out (fault mode 1) flight anomaly 110 s after mission plan initiation. This simulation shows the UAV flying to its mission endpoint because that endpoint is within the calculated FGIF. The beginning and end waypoints are assumed to have casualty expectation values of 0, which is why the UAV will always return to base or finish mission if those points are within its FGIF. This result is also reflected in Table. 4.2, where the recorded value of CE with GIHM for fault mode 1 at fault time 110 s was 0 fatalities per flight hour.

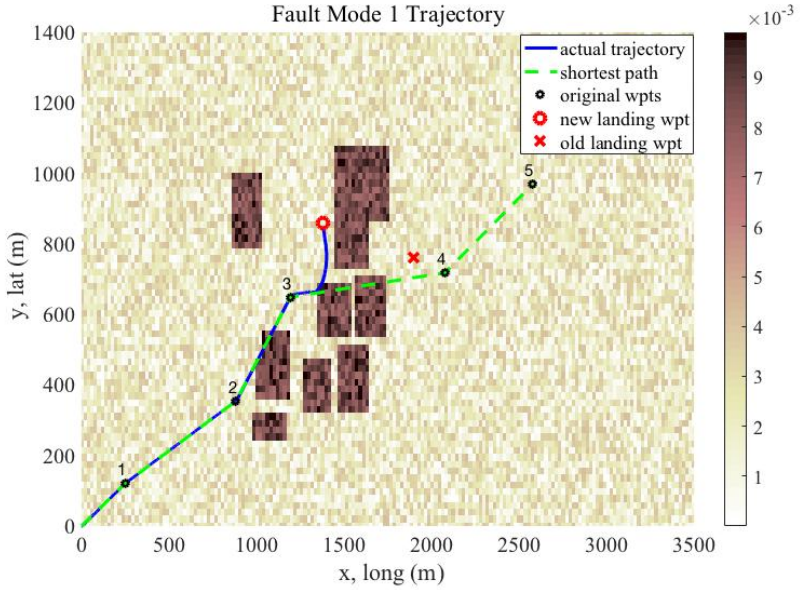


Figure 4.6: UAV trajectory for when fault mode 1(engine failure) is detected at 80 s. The aircraft starts from home at the origin and travels to the waypoints in numerical order until the fault mode is detected. "New landing waypoint" is the waypoint sent by GIHM and "old landing waypoint" is where the UAV would have landed without GIHM.

Fault mode 2 was also investigated at a time of 80 s to compare flight performance to that of the UAV experiencing fault mode 1 at a time of 80 s. Fig. 4.10 shows the simulated flight trajectory for the UAV experiencing fault mode 2

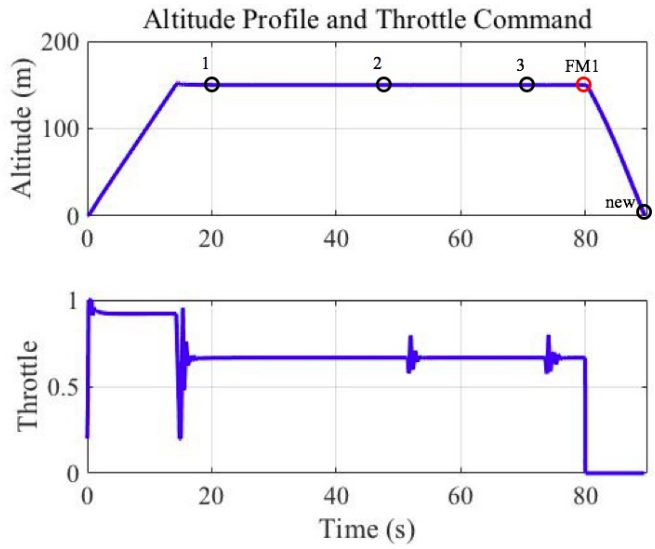


Figure 4.7: UAV altitude and throttle command profiles for when fault mode 1(engine failure) is activated at 80 s. The numbered circles show the mission waypoints, with the one labeled 'new' representing the new low hazard waypoint generated by GIHM. The red circle represents when the fault mode was detected. The aircraft starts on the ground, climbs to reach waypoint 1, then descends when the fault mode is detected at 80 s.

(engine and rudder failure) 80 s after simulation initiation. At 80 s, the UAV is just past waypoint 3, when the decision engine detects a stuck rudder control surface and engine failure. The UAV is still able to change its heading using its ailerons, as indicated by the turn executed by the UAV to reach the new minimum hazard waypoint. Fig. 4.11 shows the altitude, throttle, and rudder profiles for fault mode 2 detection at 80 s. The altitude plot shows the UAV experiences fault mode 2 at 80 s, which is why the aircraft begins its descent at 80 s. The throttle and rudder profiles show the throttle of the UAV is 0 after 80 s and the rudder is stuck at a deflection of approximately 0.1 radians. Fig. 4.12 shows the Euler angles and Euler rates for the UAV experiencing fault mode 2 80 s after flight initiation. Note the

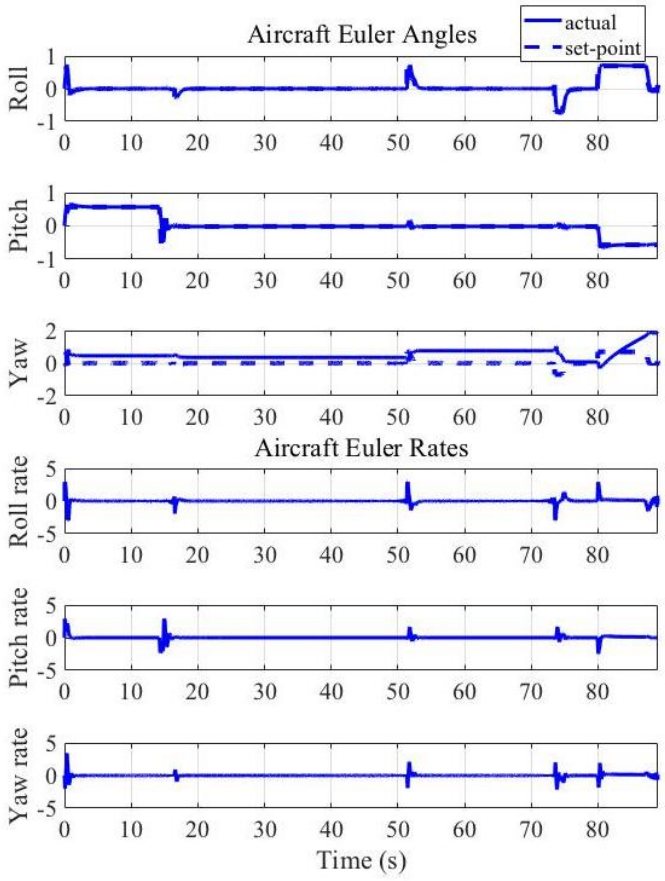


Figure 4.8: UAV Euler angles and rates profiles for when fault mode 1(engine failure) is activated at 80 s. Angles are in radians. The dashed line on the Euler angle plots show the controller commands and the solid line shows the actual aircraft states. The aircraft has drastic changes in all three Euler angles when the fault mode is detected at 80 s because of the maneuvers made by the UAV to reach the low hazard waypoint.

change in Euler angles at time 80 s, which is a result of the aircraft changing its heading to fly to the new low hazard waypoint generated by GIHM.

Next, we look at UAV performance when fault mode 4 is detected. Fig. 4.13 and Fig. 4.14 show the simulated flight trajectory and altitude profile of the UAV experiencing the engine and ailerons out (fault mode 4) flight anomaly 63 s after mission plan initiation. At 63 s, the UAV is almost at waypoint 3. At 63 s, the

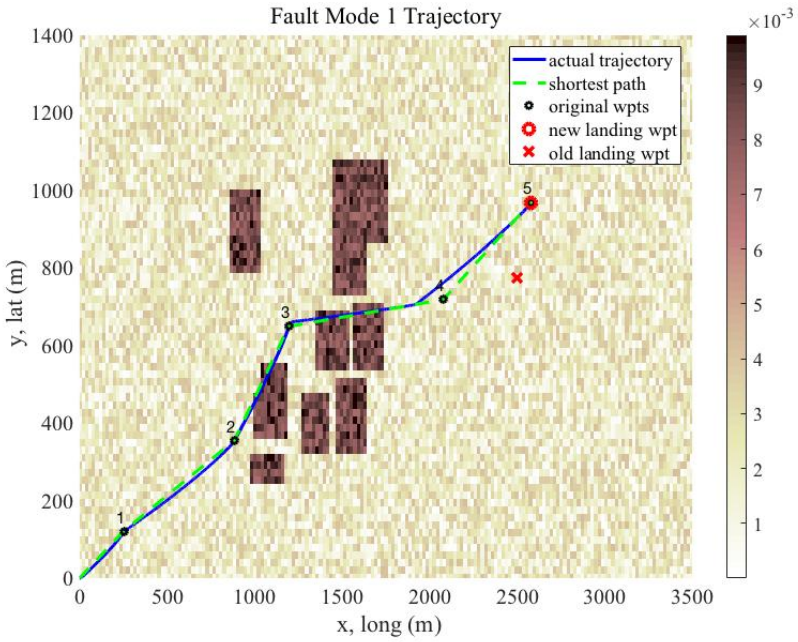


Figure 4.9: UAV trajectory for when fault mode 1(engine failure) is activated at 110 s. The aircraft starts from home at the origin and travels to the waypoints in numerical order until the fault mode is detected. "New landing waypoint" is the waypoint sent by GIHM and "old landing waypoint" is where the UAV would have landed without GIHM.

engine ailerons fault mode is detected, which results in the change in mission plan. Fig. 4.14 also shows the throttle and rudder profiles of the UAV experiencing fault mode 4 63 s after mission initiation. From the altitude and trajectory plots, it can be concluded that the UAV found a minimum ground impact point and is flying to that point. The new priority waypoint is at the same heading as the UAV when the fault mode was detected because the UAV is unable to change its heading due to the ailerons being stuck. The trajectory plot shows where the UAV would have landed compared to where it was able to maneuver to. From the heat map, the UAV is able to change its pitch angle to guide the aircraft to the ground before it

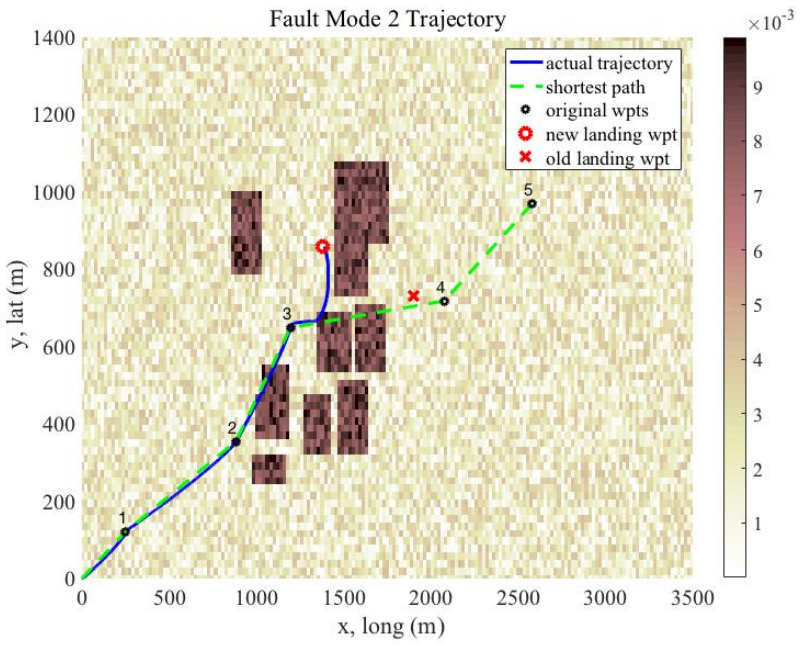


Figure 4.10: UAV trajectory for when fault mode 2 (engine and rudder failure) is activated at 80 s. The aircraft starts from home at the origin and travels to the waypoints in numerical order until the fault mode is detected. "New landing waypoint" is the waypoint sent by GIHM and "old landing waypoint" is where the UAV would have landed without GIHM.

crashes in the highly populated group of buildings, which is where the UAV would have landed without GIHM.

Fig. 4.15 shows the Euler angles and Euler rates for the UAV experiencing fault mode 4 63 s after flight initiation. Note the change in pitch angle and pitch rate at time 63 s. This is a result of the aircraft changing its orientation to descend to the new priority waypoint generated by GIHM. Notice that the roll and yaw angles remain unchanged after the fault mode is detected. This is because the UAV cannot change its yaw or roll angles because of its malfunctioning ailerons. Simulation

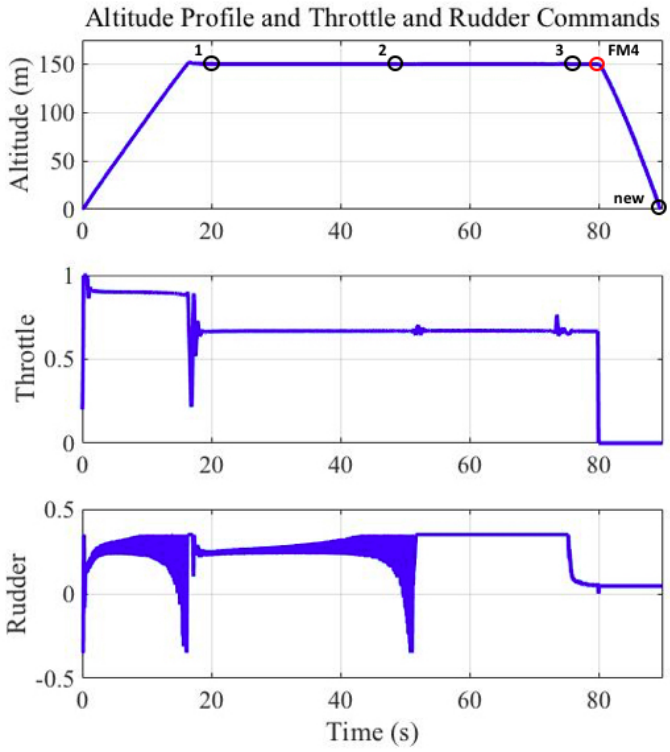


Figure 4.11: UAV altitude, throttle, and rudder command profiles for when fault mode 2 (engine and rudder failure) is detected at 80 s. The numbered circles show the mission waypoints, with the one labeled 'new' representing the new landing waypoint generated by GIHM. The red circle represents when the fault mode was detected. The aircraft starts on the ground, climbs to reach waypoint 1, then descends when the fault mode is detected at 80 s.

plots were provided only for fault modes 1, 2, and 4 because these show the most interesting results.

Computation time for the GIHM module ranges between 0.01 s and 0.04 s on a 2015 MacBook Air with 1.6 GHz I5 processor, depending on which fault mode the UAV is experiencing. Fault modes that allow for heading changes require more computation time because of the added amount of data points in the FGIF. From the plots, it is concluded that when a fault mode was detected, GIHM calculated

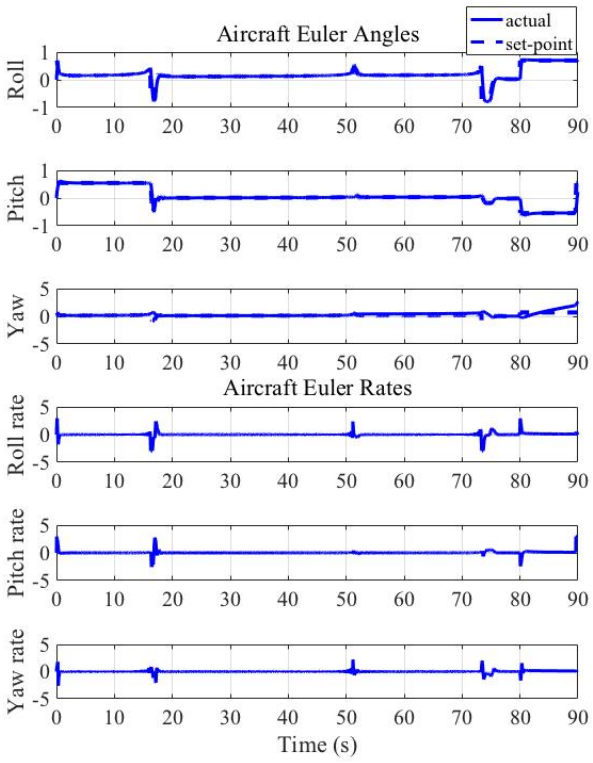


Figure 4.12: UAV Euler angles and rates profiles for when fault mode 2 (engine and rudder failure) is detected at 80 s. Angles are in radians. The aircraft has changes in all three Euler angles when the fault mode is detected at 80 s because of the maneuvers made by the UAV to reach the low hazard waypoint.

the lowest casualty expectation impact waypoint and was able to maneuver to that point. The new waypoints for all trajectory plots successfully avoided the highly populated areas, while the aircraft would have landed on the edge of the highly populated zones if the UAV had not used the GIHM module.

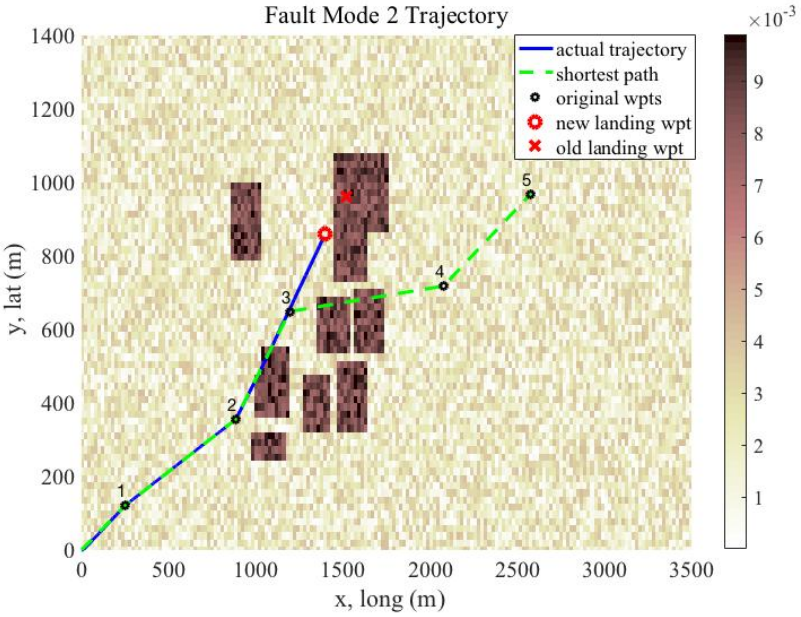


Figure 4.13: UAV trajectory for when fault mode 4 (engine and aileron failure) is detected at 63 s. The aircraft starts from home at the origin and travels to the waypoints in numerical order until the fault mode is detected. "New landing waypoint" is the waypoint sent by GIHM and "old landing waypoint" is where the UAV would have landed without GIHM.

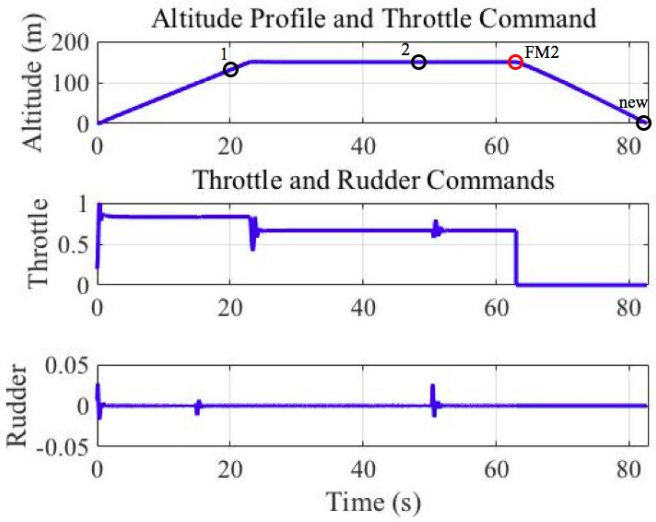


Figure 4.14: UAV altitude, throttle, and rudder command profiles for when fault mode 4 (engine and aileron failure) is detected at 63 s. The numbered circles show the mission waypoints, with the one labeled 'new' representing the new landing waypoint generated by GIHM. The red circle represents when the fault mode was detected. The aircraft starts on the ground, climbs to reach waypoint 1, then descends when the fault mode is detected at 63 s.

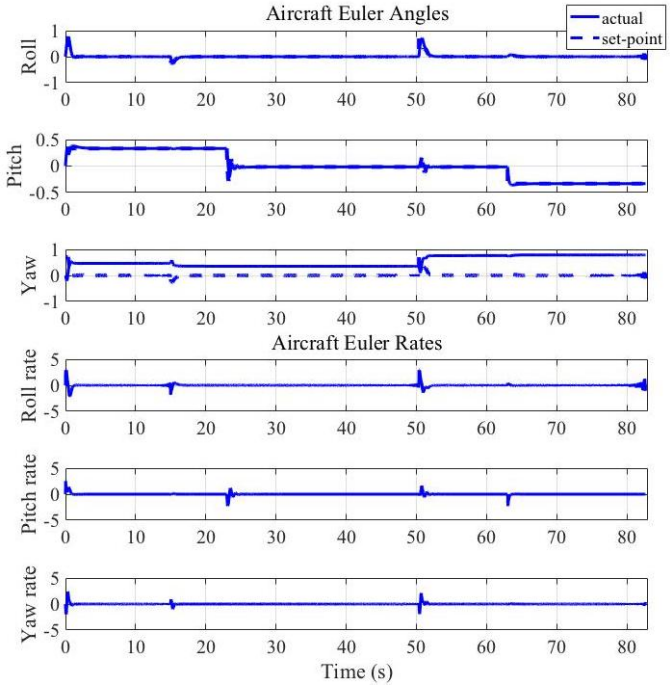


Figure 4.15: UAV Euler angles and rates profiles for when fault mode 4 (engine and aileron failure) is detected at 63 s. Angles are in radians. The aircraft has changes in pitch angle, but not roll or yaw angles when the fault mode is detected at 63 s because of the maneuvering limitations imposed by the fault mode.

Chapter 5: Conclusion

5.1 Conclusion

This work provided the architecture to combine a collision detection and avoidance module, decision engine, and ground impact hazard mitigation module, which is a big step towards developing UAV flight control software safe enough for UAVs to fly in the NAS. A preliminary decision engine was presented, which could detect engine and control surface fault modes by monitoring UAV flight states real-time. A high-level look into a collision detection and avoidance algorithm was presented to show how it will integrate with the nominal flight control software.

This work also combined a reachable ground footprint model with a precise spatial resolution population dataset, all while integrating with a flight simulation software that included a 6-DOF aircraft model, path planning, and autopilot control. Understanding the footprint for a UAV experiencing flight anomalies is an important first step in safer UAV control software. However, existing ground footprint models of UAV decision making either do not take into account population density in the reachable footprint, or the population density data accuracy needs to be improved. A precise population dataset is an important addition to reachable ground footprint models because the UAV now possesses the ability to quantify its ground impact

after determining everywhere it can land. GIHM successfully decreases the casualty expectation of a small UAV by an average of 24.718 fatalities per 100,000 flight hours, which is a 97.3% decrease in casualty expectation. The development of the four fault modes allowed for a basis of investigation into where a UAV might land given the restrictions that specific fault modes induce and how the UAV can reduce casualty expectation for those flight anomalies.

5.2 Future Work

This thesis has provided an initial foundation for which UAV technology can become safe enough to allow for their integration into the NAS. However, there is still work to be done in regards to refinement of current models and integration of all parts of SARGM with nominal UAV control software. The GIHM module has already begun the integration phase, but has more integration testing and refinement to go. Furthermore, additional fault modes need to be explored. This includes generating reachable footprints for an aircraft that has a functioning engine. A functional engine allows for use of maximum glide ratio gliding equations to generate the maximum reachable footprint. Additionally, the GIHM algorithm needs to include landing zone determination logic, as opposed to selecting only crash sites. This will require integration of a topology database with the current decision algorithm used in GIHM. LandScan USA also provides data on daytime vs. nighttime population in the USA, which could help provide more accurate collective risk profiles and assist in decision making for when to perform missions.

Future work for the collision detection and avoidance module includes integration with the nominal UAV control software, and developing and integrating sensor models to allow for generation of sensor data when simulating collision detection and avoidance functionality during integration tests. Future work for the decision engine module includes interface testing between the collision detection and avoidance and GIHM modules. Detection of fault modes from sensor data needs further development and additional integration testing is required for full implementation of the decision engine with the nominal UAV control software. Furthermore, instability recovering tactics need to be researched to execute various flight control instability mitigation maneuvers during flight anomalies before diagnosing a critical fault mode. Finally, extensive hardware testing is required for verification and validation of the system design. This includes hardware testing of the individual modules of SARGM, and the fully integrated, all-encompassing flight control software package.

Bibliography

[1] T. Cox, C. Nagy, M. Skoog, and I. Somers, *Civil UAV capability assessment*, accessed April 9, 2018. [Online]. Available: https://www.nasa.gov/centers/dryden/pdf111761main_UAV_Capabilities_Assessment.pdf

[2] 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.

[3] B. Elias, "Unmanned aircraft operations in the national airspace system," *Congressional Research Service Report for Congress*, 2012.

[4] C. Belcastro, D. Klyde, M. Logan, R. Newman, and J. Foster, "Experimental flight testing for assessing the safety of unmanned aircraft system safety-critical operations," *17th AIAA Aviation Technology, Integration, and Operations Conference*, p. 3274, 2017.

[5] D. Bruke, C. Hall, and S. Cook, "Sense and avoid in uas: research and applications," *John Wiley & Sons*, 2012.

[6] Boeing, *Statistical Summary of Commercial Jet Airplane Accidents*, accessed September 1, 2017. [Online]. Available: http://www.boeing.com/resources/boeingdotcom/company/about_bca/pdf/statsum.pdfl

[7] C. Bogdiukiewicz, M. Butler, T. Hoang, M. Paxton, J. Snook, X. Waldron, and T. Wilkinson, "Formal development of policing functions for intelligent systems," *28th International Symposium on Software Reliability Engineering*, 2017.

[8] R. Loh, Y. Bian, and T. Roe, "Safety requirements for unmanned aerial vehicles (uav) in future civil airspace," *Position, Location, And Navigation Symposium*, 2006.

[9] D. Dawei and L. Baoan, "Flight safety control and ground test on uav," *Chinese Guidance, Navigation and Control Conference*, 2014.

[10] "Unmanned aircraft systems (uas)," *International Civil Aviation Organization*, 2011.

[11] K. Valavanis, "Unmanned aircraft systems challenges in design for autonomy," *Proceedings of the 11th International Workshop on Robot Motion and Control*, 2017.

[12] C. Lum and B. Waggoner, "A risk based paradigm and model for unmanned aerial systems in the national airspace," *American Institute of Aeronautics and Astonautics*, 2011.

[13] R. S. Group, "Range safety criteria for unmanned air vehicles."

[14] D. Burke, C. Hall, and S. Cook, "System-level airworthiness tool," *Journal of Aircraft*, pp. 777–785, 2011.

[15] J. Stevenson, S. OYoung, and L. Rolland, "Estimated levels of safety for small unmanned aerial vehicles and risk mitigation strategies," *Journal of Unmanned Vehicle Systems*, pp. 205–215, 2015.

[16] R. Clothier, R. Walker, N. Fulton, and D. Campbell, "A casualty risk analysis for unmanned aerial system (uas) operations over inhabited areas," *Second Australasian Unmanned Air Vehicle Conference*, pp. 1–15, 2007.

[17] E. Cohen, J. Herrmann, and S. Azarm, "Risk-based path planning optimization methods for unmanned aerial vehicles over inhabited areas," *Journal of Computing and Information Science in Engineering*, vol. 6, June 2016.

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

[19] M. Coombes, W. Chen, and P. Render, "Reachability analysis of landing sites for forced landing of a uas," *International Conference on Unmanned Aircraft Systems (ICUS)*, 2013.

[20] P. Wu and R. Clothier, "The development of ground impact models for the analysis of the risks associated with unmanned aircraft operations over inhabited areas," *Eleventh Probabilistic Safety Assessment and Management Conference*, 2012.

[21] N. Bradley and D. Burke, "Potential crash location (plc) model," *Naval Air Warfare Center Aircraft Division*, 2012.

[22] E. Atkina, I. Portillo, and M. Strube, "Emergency flight planning applied to total loss of thurst," *Journal of Aircraft*, pp. 1205–1216, 2006.

[23] L. Barr, R. Newman, E. Ancel, C. Belcastro, J. Foster, J. Evans, and D. Klyde, "Preliminary risk assessment for small unmanned aircraft systems," *17th AIAA Aviation Technology, Integration, and Operations Conference*, 2017.

[24] B. Bhaduri, E. Bright, P. Coleman, and M. Urban, "Landscan usa: a high-resolution geospatial and temporal modeling approach for population distribution and dynamics," *GeoJournal*, vol. 69, pp. 103–117, 2007.

[25] X. Li, R. Rowley, J. Kostelnick, D. Braaten, J. Meisel, and K. Hulbutta, "Gis analysis of global impacts from sea level rise," *Photogrammetric Engineering and Remote Sensing*, vol. 75, no. 7, pp. 807–818, 2009.

[26] P. Sutton, C. Elvidge, and T. Obremski, "Building and evaluating models to estimate ambient population density," *Photogrammetric Engineering and Remote Sensing*, vol. 69, no. 5, pp. 545–553, 2003.

[27] C. Linard and A. Tatem, "Large-scale spatial population databases in infectious disease research," *International journal of health geographics*, 2012.

[28] J. H. University, *Systems context diagrams*, accessed March 12, 2018. [Online]. Available: https://ep.jhu.edu/about-us/news-and-media/systems-context-diagrams

[29] AcqNotes, *Requirements development*, accessed March 12, 2018. [Online]. Available: http://acqnotes.com/acqnote/tasks/measures-of-effectivenessrequirements

[30] M. Verhaegen, S. Kanev, R. Hallouzi, C. Jones, J. Maciejowski, and H. Smail, "Fault tolerant flight control-a survey," *Fault tolerant flight control*, pp. 47–89, 2010.

[31] K. Williams, "A summary of unmanned aircraft accident/incident data: human factors implications," *FAA Civil Aerospace Medical Institute*.

[32] N. G. R. Center, *The Drag Coefficient*, accessed February 2, 2018. [Online]. Available: https://www.grc.nasa.gov/www/k-12/airplane/dragco.html

[33] C. Belcastro, D. Klyde, M. Logan, R. Newman, and J. Foster, "Experimental flight testing for assessing the safety of unmanned aircraft system safety-critical operations," *17th AIAA Aviation Technology, Integration, and Operations Conference*, p. 3274, 2017.

[34] S. Wolf, "Modeling small unmanned aerial system mishaps using logistic regression and artificial neural networks," *Air Force Institute of Technology*, 2012.

[35] NASA, *Glide angle and glide ratio*, accessed August 3, 2017. [Online]. Available: https://www.grc.nasa.gov/www/k-12/airplane/glidang.html