ABSTRACT

Title of Dissertation:	CERTIFYING AN AUTONOMOUS SYSTEM TO COMPLETE TASKS CURRENTLY RESERVED FOR QUALIFIED PILOTS
	Donald H. Costello III Doctor of Philosophy, 2020
Dissertation Directed by:	Assistant Professor Huan Xu Department of Aerospace Engineering

When naval certification officials issue a safety of flight clearance, they are certifying that when the vehicle is used by a qualified pilot they can safety accomplish their mission. The pilot is ultimately responsible for the vehicle. While the naval safety of flight clearance process is an engineering based risk mitigation process, the qualification process for military pilots is largely a trust process. When a commanding officer designates a pilot as being fully qualified, they are placing their trust in the pilot's decision making abilities during off nominal conditions. The advent of autonomous systems will shift this established paradigm as there will no longer be a human in the loop who is responsible for the vehicle. Yet, a method for certifying an autonomous vehicle to make decisions currently reserved for qualified pilots does not exist. We propose and exercise a methodology for certifying an autonomous system to complete tasks currently reserved for qualified pilots. First, we decompose the steps currently taken by qualified pilots to the basic requirements. We then develop a specification which defines the envelope where a system can exhibit autonomous

behavior. Following a formal methods approach to analyzing the specification, we developed a protocol that software developers can use to ensure the vehicle will remain within the clearance envelope when operating autonomously. Second, we analyze flight test data of an autonomous system completing a task currently reserved for qualified pilots while focusing on legacy test and evaluation methods to determine suitability for obtaining a certification. We found that the system could complete the task under controlled conditions. However, when faced with conditions that were not anticipated (situations where a pilot uses their judgment) the vehicle was unable to complete the task. Third, we highlight an issue with the use of onboard sensors to build the situational awareness of an autonomous system. As those sensors degrade, a point exists where the situational awareness provided is insufficient for sound aeronautical decisions. We demonstrate (through modeling and simulation) an objective measure for adequate situational awareness (subjective end) to complete a task currently reserved for qualified pilots.

CERTIFYING AN AUTONOMOUS SYSTEM TO COMPLETE TASKS CURRENTLY RESERVED FOR QUALIFIED PILOTS

by

Donald H. Costello III

Dissertation 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
Doctor of Philosophy
2020

Advisory Committee:
Assistant Professor Huan Xu, Chair/Advisor
Professor Adam Porter, Dean's Representative
Professor Jeffrey Herrmann
Professor Miao Yu
Associate Professor Sarah Bergbreiter

Dedication

To my wife, for her love and support.

To my children, for inspiring me.

To my mother, for always beveling in me.

To my father, who I will always look up to.

Acknowledgments

I owe my gratitude to all the people who have made this thesis possible and because of whom my graduate experience has been one that I will cherish forever. There is no way I will be able to thank everyone as literally hundreds of individuals have assisted in this effort.

First and foremost I'd like to thank my wife, Jenna Costello, for her love and support during the last 20 years. Through the deployments and forced separations due to my military career she has been the rock that has kept our family together. Words cannot express the gratitude I owe you.

Second, I would like to thank all of the individuals from the Navy that have made this work possible. Two in particular had a dramatic impact on this effort. Mr. John O'Connor for keeping me in mind as various opportunities came up to further my education and for always having an open door for an old shipmate. Mr. Steven Kracinovich for helping further my education through the NAWCAD Center for Autonomy.

I would also like to acknowledge the help and support I received from some technical experts during my research. Dr. Joseph Slagel from the NASA Langley Formal Methods group for his help in focusing my efforts in the area of formal methods. Mr. Jason Jewell, from Auora Flight Sciences for his expertise and the support in furthering autonomy in aviation through the various test programs they are accomplishing. Dr. Nicholas Hanlon from AFRL and Mr. Brian Lucas NAWCAD for their support in working in the AFSIM M&S environment.

Next I would like to thank the various support personnel at the University of Maryland for working with me through all of the wickets of the PhD process. As I was not your typical PhD student, but a full time naval officer (who ended up circling the world during a military deployment during my candidacy). Mr. Otto Fandino and Ms. Kerri James were always available to answer an email or a phone call to assist me through the process. Dr. Lina Castano for working with me on LaTeX and being another set of eyes on my research. As I have spent the majority of the last 20 years outside of the academic environment, her efforts and those of my adviser were critical on framing my research in such a way to allow me to complete this academic course of study.

I would also like to thank my first academic adviser from the United States Naval Academy. In the fall of 1998 Dr. Brad Bishop inspired a young Midshipman to never give up and continue his education once he was in the fleet.

Finally, I want to thank my adviser, Assistant Professor Huan Xu, for working with me on a topic of interest to the United States Navy. In 2017 I was looking for an adviser to help guide my research as a non-standard student. She agreed and has always been there to answer questions and point me in the right direction. It has been a pleasure to work with and learn from such an extraordinary individual.

Please note: An Editor was not used in the preparation of this dissertation.

Table of Contents

Dedication ii

Acknowledgements iii

Table of Contents v

List of Tables ix

List of Figures xi

List of Abbreviations xv

Chapter 1: Introduction 1
- 1.1 Motivation . . . 5
- 1.2 Problem Formulation . . . 7
- 1.3 Background . . . 8
- 1.4 Certification Methodology . . . 12
- 1.5 Contributions . . . 14
- 1.6 Dissertation Outline . . . 16

Chapter 2: Literature Review 18
- 2.1 Overview . . . 18
- 2.2 Review of Automation within Tactical Naval Aviation . . . 19
  - 2.2.1 Autopilot/Pilot Relief Modes in the F/A-18 Hornet Family of Aircraft . . . 19
  - 2.2.2 Automated Landing of Tactical Jets on Aircraft Carriers . . . 21
  - 2.2.3 X-47 CVN Demonstrator . . . 25
- 2.3 Naval Certification Processes . . . 28
  - 2.3.1 Naval Aviation Aircraft Certification Process . . . 28
  - 2.3.2 Naval Aviator Certification Process . . . 30
  - 2.3.3 NAVAIR 4.0P Certification Process . . . 32
- 2.4 How to Build Trust in Autonomous Systems Leading to Certification 37
- 2.5 Bringing Autonomy to Military Aviation . . . 42
  - 2.5.1 Challenges . . . 43
  - 2.5.2 Gaps . . . 45
  - 2.5.3 Goals . . . 47

2.5.4 Software Development and its Implications on Certification of Autonomy within Naval Aviation 50
2.6 Formal Methods Relation to Naval Aviation Automation 57
2.6.1 Model Checking 58
2.6.2 Theorem Provers 63
2.6.3 Run Time Assurance 67
2.7 Other Topics for Autonomy in Military Aviation 70
2.7.1 Requirements and metrics 70
2.7.2 Normative Oracle Generation 72
2.7.3 Coactive Design 73
2.7.4 Implication of Learning Autonomous Systems 75
2.7.5 Modeling and Simulation Considerations 77
2.7.6 FAA See and Avoid Research for Autonomy 81
2.7.7 Naval See and Avoid Certification 82
2.7.8 AACUS 84
2.8 Helicopter Landing Mission Overview 87
2.8.1 Confined Area Landing (CAL)/Landing Zone (LZ) 89
2.8.2 SWEEP 90
2.9 Helicopter Aircraft Command (HAC) Qualification 93
2.9.1 Helicopter Second Pilot 94
2.9.2 HAC Syllabus 96
2.9.3 HAC Requirements 96
2.9.4 HAC Oral Board 97
2.9.5 Commanders Trust 99

Chapter 3: Requirements Definition 101
3.1 Naval Aviation Certification Processes 104
3.1.1 Current Certification Process for Naval Aircraft/Systems . . . 104
3.1.2 Current Certification Process for Helicopter Aircraft Commander (HAC) 106
3.2 Requirements Definition and The Specification 107
3.2.1 Development of the Basic Requirements 108
3.2.2 Specification 110
3.3 Analysis of the Specification 113
3.3.1 Operational Procedure Table 114
3.3.2 Consistency and Completeness 117
3.3.3 Theorem Proving Model 117
3.3.4 PVS Model 118
3.4 Protocol 122
3.4.1 Evidence Leading to a Naval Flight Clearance 125
3.5 Chapter Summary 128

Chapter 4: Flight Test of an Autonomous System 130
4.0.1 Current Methods for Flight Certification 134
4.1 Certifying Autonomy for the CAL/LZ Mission 135

4.1.1 Flight Test Overview . . . 137
4.1.2 System Under Test (AACUS/TALOS) Overview . . . 140
4.2 Developmental Flight Test of AACUS/TALOS . . . 142
4.2.1 Requirements of AACUS/TALOS for the Autonomous CAL/LZ Mission . . . 142
4.2.2 Developmental Flight Test Matrix . . . 145
4.2.3 Summary of Developmental Flight Test Events . . . 150
4.2.4 DT Results and DT/OT Transition Recommendation . . . 155
4.3 Operational Flight Test of AACUS/TALOS . . . 156
4.3.1 Goals and Expectations of the System in OT . . . 157
4.3.2 Operational Flight Test Matrix . . . 158
4.3.3 Summary of Operational Test Events . . . 159
4.3.4 AACUS/TALOS System Suitability . . . 163
4.4 Analysis of the Test Results as it Relates to Certifying Autonomy . . . 165
4.4.1 Insufficient SA that May have Led to a Mishap in an Autonomous Vehicle (Outside of the CAL/LZ Mission) . . . 168
4.5 Chapter Summary . . . 172

Chapter 5: Developing a Objective Measure for a Subjective End 175
5.1 Overview of SA and Developing a Objective Measure for a Subjective End . . . 178
5.1.1 Current Methods for Flight Certification and SA . . . 179
5.1.2 Development of a Objective Measure (Cooper-Harper Scale) for a Subjective End (Handling Qualities) . . . 183
5.1.3 Cooper-Harper Adjusted for Confidence in Automation . . . 184
5.2 Problem Formation . . . 188
5.2.1 M&S Environment and Statement of the Problem . . . 189
5.2.2 Experiment Factors (Variables) . . . 191
5.2.3 Measures of Performance (MOP) . . . 193
5.2.4 Fused Error Distance as a Function of Sensor Error . . . 194
5.3 Experimental Results and Analysis . . . 196
5.3.1 Conduct of the Experiment . . . 197
5.3.2 Analysis of the Data . . . 199
5.3.3 DOE Conclusions . . . 202
5.4 Chapter Conclusion . . . 205

Chapter 6: Conclusions 206
6.1 Summary . . . 206
6.2 Original Contributions . . . 209
6.3 Outlook for Future Work . . . 211

Appendix A:Experimental Results 213
A.1 All Possible Outcomes of the Eight Protocol Assessments Truncated in Table 3.2 . . . 213

A.2 All 729 Combinations of the Six Variables (error σs) Truncated in Table 5.4 227

Bibliography 267

List of Tables

1.1 Technology Readiness Levels Summary Used by NASA [14] 5

3.1 Event Description for the State Machine Specification Which Details the Decision Process for a Unmanned System to Make a Decision Currently Reserved for a Qualified Pilot 112
3.2 Depiction of 5 of the 256 Possible Outcomes of the 8 Protocol Assessments 127

4.1 Completed DT Test Matrix of AACUS/TALOS for the Autonomous CAL/LZ Mission (P = Pass, F = Fail, N/A = Not Applicable) 150
4.2 Legend for Colors in TALOS LZ Interpretation [132] 152
4.3 Completed OT Flight Test Matrix of AACUS/TALOS for the Autonomous CAL/LZ Mission 160

5.1 Summary of the Terms in the Multiple Regression Model (Equation 5.1). 196
5.2 Summary of the Terms in the Predictive Equations (Equation 5.2 and 5.3). 198
5.3 10 of the 729 Data Points. Y_{20} is the 20 nm Error Distance in Meters. Y_{10} is the 10 nm Error Distance in Meters. X_1 is the Value of One σ Error in IRST Azimuth in Degrees. X_2 is the Value of One σ Error in IRST Elevation in Degrees. X_3 is the Value of One σ Error in IRST Range in nm. X_4 is the Value of One σ Error in Radar Azimuth in Degrees. X_5 is the Value of One σ Error in Radar Elevation in Degrees. X_6 is the Value of One σ Error in Radar Range in nm. . . 198
5.4 10 of the 729 Data Points. Y_{20} is the 20 nm Error Distance in Meters. Y_{10} is the 10 nm Error Distance in Meters. X_1 is the Value of One σ Error in IRST Azimuth in Degrees. X_2 is the Value of One σ Error in IRST Elevation in Degrees. X_3 is the Value of One σ Error in IRST Range in nm. X_4 is the Value of One σ Error in Radar Azimuth in Degrees. X_5 is the Value of One σ Error in Radar Elevation in Degrees. X_6 is the Value of One σ Error in Radar Range in nm. . . 199
5.5 20 and 10 nm Regression Data Obtained Through Microsoft Excel Multiple Regression Analysis. 200

5.6 Results From 25 Test Runs of Randomly Generated σ Values. Y_{20} is the 20 nm Error Distance in Meters. Y_{10} is the 10 mn Error Distance in Meters. X_1 is the Value of One σ Error in IRST Azimuth in Degrees. X_2 is the Value of One σ Error in IRST Elevation in Degrees. X_3 is the Value of One σ Error in IRST Range in nm. X_4 is the Value of One σ Error in Radar Azimuth in Degrees. X_5 is the Value of One σ Error in Radar Elevation in Degrees. X_6 is the Value of One σ Error in Radar Range in nm. 203
5.7 Results From 25 Test Runs Y_x, the Corresponding Results From of Predictive Equations $\hat{Y}_x$, the Absolute Distance Between the Two in Meters and Percentage. The Error Percentages are also Summarized. 204

A.1 256 Possible Outcomes of the 8 Protocol Assessments Truncated in Table 3.2 213
A.2 Results of the 729 Simulations Truncated in Table 5.4. Y_{20} is the 20 nm Error Distance in Meters. Y_{10} is the 10 nm Error Distance in Meters. X_1 is the Value of One σ Error in IRST Azimuth in Degrees. X_2 is the Value of One σ Error in IRST Elevation in Degrees. X_3 is the Value of One σ Error in IRST Range in nm. X_4 is the Value of One σ Error in Radar Azimuth in Degrees. X_5 is the Value of One σ error in Radar Elevation in Degrees. X_6 is the Value of One σ Error in Radar Range in nm [180]. 228

List of Figures

2.1 EA-18G Growler During Flight Test in 2009 [83] 20
2.2 Test Matrix for CVN PALS Certification [86] 23
2.3 Graphical Depiction of the Ideal Hook Touch Down Point for PALS Approach on a Nimitz Class Aircraft Carrier [87] 24
2.4 Graphical Depiction of the Difference Between the Beacon and Hook Flight Path on PALS Approach [86] 24
2.5 Extract From the DMOT Portion of the PALS Certification Test Plan Detailing the Requirements [86] 26
2.6 First Landing of a Large Fixed Wing Drone on Board an Aircraft Carrier, 10 July 2013 [89] 27
2.7 Three-Layered Framework for Conceptualizing Trust Variability as Presented in Reference [95] 40
2.8 Parado Airfield in Chino Hills, CA, Testing Area ("Google Earth" Image) Used in Reference [66] 48
2.9 Post Processed 3D Map of the Test Area Used in Reference [66] . . . 49
2.10 Classic V Development Cycle [8] 50
2.11 Autonomy TEVV Process Model, Integrated with the Traditional [8] 51
2.12 Typical Hybrid Autonomous System Architecture – with Suitable Analysis Techniques Noted in Reference [24] 52
2.13 V-Model as Described in Reference [20] 53
2.14 Increasing Aircraft Functionality Provided by Software [21] 54
2.15 Important Development Phases According to DO-178C, Including the Necessary Verification Steps (Architecture Design and Verification have Been Omitted) [26] 55
2.16 Run Time Assurance Architecture as Described in Reference [36] . . . 67
2.17 Goal of Decision-Making: Identify Safe Trajectory [108] 72
2.18 V-Cycle Stages for MBD-Based V&V [22] 79
2.19 Two-Stage V&V Process [63] 80
2.20 MBCS V&V Framework Process Flow and Elements [63] 81
2.21 The Total Minimum Detection Range, d_{MDR}, Needed and a Representation of the CPA [119] 83
2.22 RRT and RRT* Path Planning. Image (a) is an Example of RRT, Which has been Widely used for Path Planning of Autonomous Robotic Path Planning. Image(b) is a Midpoint of the RRT* Algorithm Where it is Defining a More Optimal Path Through the Obstacles. Image (c) is the Final Product Where the RRT* Algorithm has Defined the Optimal Path Around the Obstacles [122] 87

3.1 State Machine Specification Which Details the Decision Process for a Unmanned System to Make a Decision Currently Reserved for a Qualified Pilot 112
3.2 Operational Procedure Table Converting the State Machine Specification Into the Various Tasks Required for an Unmanned System to Make a Decision Currently Reserved for a Qualified Pilot 115
3.3 PVS Specification for SWEEP Checks to Landing Detailing the Decision Process for a Unmanned System to Make a Decision Currently Reserved for a Qualified Pilot 120
3.4 LEMMA 3 deals with the Environmental Conditions of the LZ: If the Elevation or the Winds are out of Limits the LZ is not Valid Due to Bad Conditions 121
3.5 Depiction of 11 Hypothetical LZs Against the Propositions Listed in Section 3.3.3 and Later Detailed in the PVS Model 122
3.6 Protocol Which Meets the Requirements of the Specification Detailing the Decision Process for a Unmanned System to Make a Decision Currently Reserved for a Qualified Pilot 125

4.1 Simplified Flowchart Detailing the Steps Leading to a Safety of Flight Clearance for an Autonomous System to Accomplish the CAL/LZ Mission. This Chapter Focuses on Steps 5 and 6 133
4.2 Photo of a Marine Carrying a 24x20x16 in Pelican Case During the AACUS ONR Final Demonstration [138] 145
4.3 Pilots Perspective of Two LZs Taken From a UH-1 at 200 ft AGL Over the Turf Training Area of NAS Patuxent River [139] 147
4.4 Pilots Perspective of Two LZs that Would have Been Valid if the Vehicles Were not Present, Taken from a UH-1 at 200 ft AGL Over the Turf Training Area of NAS Patuxent River [139] 148
4.5 Pilots Perspective of Surveyed LZ Used for Slope Landing Evaluation Taken from a UH-1 at 200 ft AGL Over the Turf Training Area of NAS Patuxent River [140] 149
4.6 TALOS LZ Interpretation from 410, 220 and 116 Meters During Flight 59F097. As the Vehicle Approaches the LZ its Interpretation Become Clearer. [132] 151
4.7 Two Images Relating to an Autonomous Landing in a Field During Flight 59F098 Left: TALOS Interpretation of the LZ [132] Right: Picture of the Test Vehicle Shortly After Completing an Autonomous Landing in the LZ Pictured on the Left [141] 152
4.8 Three Images Relating to an Autonomous Landing in a Simulated FOB During Flight 59F098 Top Left: TALOS Interpretation of the LZ [132] Top Right: Picture of the LZ from Ground Level [141] Bottom: AACUS/TALOS Completing an Autonomous Landing in the Simulated FOB [141] 153

4.9 TALOS Interpretation of an LZ Before (Left Image) and After (Right Image) a Golf Cart if Driven Into it Testing the Wave Off Functionality on Flight 59F096 [132] . . . 154
4.10 Two Images Relating to an LZ During Flight 59F111. Left: Google Earth Image. Right: TALOS Interpretation of the Same Location. TALOS Declared the Location Unsuitable, the Safety Pilot Disagreed [132]. . . . 161
4.11 System Under Test Performing an Autonomous Landing During Full Brownout Conditions at Twentynine Palms Marine Base During Flight 59F120 [135] . . . 162
4.12 Two Images from the System Under Test Taken at Roughly the Same Point During the Scripted Demonstration. The Left Image was Taken on 12 December 2017 and Depicts the System Approaching a Position Where the Safety Pilot Felt May have Been Unsafe (Close to the Trees). The Right Image was Taken on 13 December 2017 at Roughly the Same Place. However, in this Case the SA of the System Under Test Matched Reality as it had Climbed to a Safe Height to Avoid the Trees [132]. . . . 170
4.13 Image From the System Under Test During the 23 January 2018 Flight. At the Depicted Point During the Flight the Systems SA of the Environment Didn't Match Reality. The Autonomous Functionality would have Flown Through Some Trees and Caused a Mishap if the Safety Pilot Didn't Take Control of the Vehicle [132]. . . . 171

5.1 Cooper-Harper Rating Scale (Card Used by Handling Qualities Engineers and Test Pilots) [170, 172] . . . 185
5.2 Relating Short Period Aircraft Dynamics to Aircraft Handling Qualities Levels During Nonterminal Flight Phases that are Normally Accomplished Using Gradual Maneuvers and Without Precision Tracking, From Mil-F-8785C [171] . . . 186
5.3 PALS/Pilot Quality Rating Scale, Allows Test Pilots to Objectively Gage their Confidence in the System Under Test. Developed for PALS Testing [2, 86], and it has Been used for Evaluation of Autonomous Systems [136] . . . 187
5.4 Graphical Depiction of the the Three Possible Error Parameters of the Sensors Installed on the Bucket Fighter [179]. . . . 192

5.5 Two Screen Captures From the M&S Environment Depicting the Threat Location Based on the IRST (Red), Radar (Green), and Fused Track (White Triangle). The Threat Aircraft is Approximately 20 nm from the Bucket Fighter (UAV in the East). The Image to the Left is a View From the South and Slightly Elevated from the Engagement. The Image on the Right Depicts the Engagement From an Elevated Position in the East. The Error σ Values Were: IRST Azimuth: 9 Degrees, IRST Elevation: 3 Degrees, IRST Range: 9 nm, Radar Azimuth: 3 Degrees, Radar Elevation: 3 Degrees, Radar Range: 9 nm [180] . . . 193
5.6 Screen Capture From the Start of a Test Run. The Threat Aircraft is in the West and the Bucket Fighter (UAV) is in the East [180]. . . 195

List of Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
&	And
A	Attack
AACUS	Autonomous Aerial Cargo Utility System
Abt	Aborted
A/C	Aircraft
AC	Aircraft Circular
ACCoRD	Airborne Coordinated Conflict Resolution and Detection
ACLS	Automatic Carrier Landing System
ACP	Allied Communications Publications
ADS-B	Automatic Dependent Surveillance-Broadcast
AFIT	Air Force Institute of Technology
AFRL	Air Force Research Laboratory
AFS	Aurora Flight Sciences
AFSIM	Advanced Framework for Simulation, Integration and Modeling
AGL	Above Ground Level
AI	Artificial Intelligence
ANOVA	Analysis of Variation
AOA	Angle of Attack
ASD(R&E)	Assistant Secretary of Defense, Research and Engineering
ATC	Auto Throttle Control
ATP	Allied Tactical Publication
AVO	Air Vehicle Operator
B	Bomber
BCN	Beacon
CA	California
CAL	Coffined Area Landing
CCMT	Cell to Cell Mapping Technique
Cert.	Certification
CG	Center of Gravity

CHR	Cooper-Harper Rating
CNAF	Commander Naval Air Forces
CNATRA	Chief of Naval Air Training
CO	Commanding Officer
COTS	Commercial Off The Shelf
CPA	Closest Point of Approach
CVN	Aircraft Carrier, Fixed Wing, Nuclear
DAIR	Direct Altitude and Identity Readout
DARPA	Defense Advanced Research Projects Agency
DFM	Dynamic Flowgraph Methodology
d_{MDR}	Minimum Detection Range with the Slack Parameter Safety Factor
DMOT	Detailed Method of Test
DO	Document
DoD	Department of Defense
DOE	Design of Experiments
DOF	Degree of Freedom
DoT	Department of Transportation
DT	Developmental Test
EA	Electronic Attack
EO	Electro-Optical
EPNER	Ecole de Personnel Navigant D'Essais et de Reception
Ext.	Extension
F	Fighter
F/A	Fighter/Attack
FAA	Federal Aviation Administration
FAR	Federal Aviation Regulation
FOB	Forward Operating Base
FSM	Finite State Machine
ft	Feet
FXP	Fleet Exercise Publication
EMD	Engineering and Manufacturing Development
Equ.	Equation
GN&C	Guidance, Navigation, and Control
GPS	Global Positioning System
GSN	Goal Structuring Notation

H	Helicopter
H/W	Hardware
HAC	Helicopter Aircraft Commander
HFS	Hierarchical Functional Specification
HIGE	Hover in Ground Effect
HIL	Hardware-in-the-Loop
HILS	Hardware-in-the-Loop Simulation
HOGE	Hover Out of Ground Effect
HQ	Handling qualities
HSC	Helicopter Sea Combat Squadron
HSM	Helicopter Maritime Strike Squadron
HVAA	High Value Airborne Asset
IAW	In Accordance With
ICAO	International Civil Aviation Organization
ICLS	Instrument Carrier landing System
IDA	Institute for Defense Analysis
IFLOS	Improved Fresnel Lens Optical System
in	Inch
INS	Inertial Navigation System
Int.	Intercept
IR	InfraRed
IRAD	Independent Research and Development
ISR	Intelligence, Surveillance and Reconnaissance
ITP	Industrial Test Procedure
ITX	Integrated Training Exercise
JANAP	Joint Army, Navy, Air Force Publication
JPS	Java PathFinder
kts	Nautical Miles Per Hour
LCDR	Lieutenant Commander
Lds	Landings
LHA	Helicopter-Carrying Amphibious Assault Ship
LHD	Amphibious Assault Ship (Multipurpose)
LiDAR	Light Detection and Ranging
LT	Lieutenant
LTJG	Lieutenant Junior Grade

LZ	Landing Zone
m	Meters
M&S	Modeling and Simulation
MATLAB	Matrix Laboratory
MAV	Micro Aerial Vehicle
MBCS	Model-Based Control System
MBD	Model Based Development
MCAS	Marine Corps Air Station
MGRS	Military Grid Reference System
Mil	Military
MIT	Massachusetts Institute of Technology
MOP	Measures of Performance
MQ	Multi-Mission Unmanned Aerial Vehicle
MSL	Mean Sea Level
MUM-T	Manned-Unmanned Teaming
N2	Engine Core (High Pressure Compressor) Speed in RPM
NAS	Naval Air Station
NASA	National Aeronautics and Space Administration
NATOPS	Naval Air Training and Operating Procedures Standardization
NAVAIR	Naval Air Systems Command
NAVAIRSYSCOM	Naval Air Systems Command
NAWCAD	Naval Air Warfare Center Aircraft Division
NFO	Naval Flight Officer
nm	Nautical Miles
NWP	Naval Warfare Publication
O	Officer
O&M	Operations and Maintenance
Obst	Obstruction
ONR	Office of Naval Research
OODA	Observe, Orient, Decide, and Act
OPNAV	Office of the Chief of Naval Operations
OT	Operational Test
PALS	Precision Approach Landing System
PCL	Pocket Checklist
PIC	Pilot in Command
PMA	Program Office

PQR	PALS/Pilot Quality Rating
PV	Pressure * Volume
PVS	Prototype Verification System
R	Reliability
Radar	Radio Detection and Ranging
RC	Remote Controlled
RAG	Replacement Air Group
RDT&E	Research, Development, Test & Evaluation
Ref.	Reference
Reqmts	Requirements
RESET	Return to station after a RETROGRADE
RETROGRADE	Withdraw from station in respionse to a threat
RH-RRT*	Receding Horizon-Based RRT*
ROME	Run-time Observation-based Margin Estimation
RQ	Reconnaissance Unmanned Aerial Vehicle
RRT	Rapidly Exploring Random Tree
RT	Ideal Gas Constant * Temperature
RTA	Run Time Assurance
RTB	Return to Base
S	Anti-Submarine Warfare
S&T	Science and Technology
S/W	Software
SA	Situational Awareness
SAR	Search and Rescue
SCATANA	Security Control of Air Traffic and Air Navigation Aids
SCRAM	Egress for defensive or survival reasons
SD	Standard Deviation
SH	Anti-Submarine Warfare Helicopter
SME	Subject Matter Expert
SOF	Safety of Flight
Specs.	Specifications
SWEEP	Size/Slope, Wind, Elevation, Escape Route, Power
SYA	Synthetic Basis
T&E	Test and Evaluation
TAE	Technical Area Experts
TALOS	Tactical Autonomous Aerial Logistics System
TCAS	Traffic Collision Avoidance System

TDP	Touchdown Point
TEVV	Test and Evaluation, Verification and Validation
TPS	Test Pilot School
Turf	Terrain Flight
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
UH	Utility Helicopter
US	United States
USAF	United States Air Force
USAFA	United States Air Force Academy
USAFTPS	United States Air Force Test Pilot School
USMC	United States Marine Corps
USN	United States Navy
USNTPS	United States Naval Test Pilot School
V&V	Verification and Validation
Ver.	Version
VIP	Very Important Person
VX	Air Test and Evaluation Squadron
W/O	Waveoff
WOD	Wind Over Deck
X	Experimental
XO	Executive Officer

Chapter 1: Introduction

Current Safety of Flight (SOF) certification standards (both military and civilian) require a qualified pilot (or operator for unmanned systems) in the loop for operation. The pilot, who controls the vehicle and makes decisions, is ultimately responsible for the safe operations of the vehicle [1]. Many modern aircraft can, and are, operated through a set of pilot relief modes (i.e., autopilots) that allow the aircraft to complete nearly the entire flight without a pilot touching the controls (which includes landing high performance jet aircraft on the pitching deck of an aircraft carrier [2]). However, the Pilot in Command (PIC) still has the responsibility for the aircraft and is expected to operate the vehicle safety under current certification standards. Federal Aviation Administration (FAA) certification for unmanned vehicles only deals with small vehicles (referred to as quad-copters or similar small drones), and requires the operator to be within line of sight of the vehicle [3]. The use of unmanned aircraft is expected to increase over the next decade as they have the capability to operate far beyond the limits of human endurance [4]. Future systems are expected to allow vehicles to operate autonomously, without an operator in the loop. They will ultimately require a new process for certifying an autonomous vehicle to accomplish tasks that are currently reserved for qualified pilots [1, 5–7].

All modern aircraft have some level of automation, and this automation is thoroughly tested during the SOF certification process. In this dissertation, a distinction has been made between automation (such as a pilot relief mode or autopilot) and autonomy. For automation, a system functions with little or no human operator involvement. However, the system performance is limited to the specific actions it has been designed to do. Typically these are well-defined tasks that have predetermined responses (such as "maintain altitude" or "fly the published approach for the duty runway"). For autonomy, a system has a set of intelligence-based capabilities that allow it to respond to situations that were not pre-programmed or anticipated (i.e., decision-based responses) prior to system deployment. Autonomous systems have a degree of self-government and self-directed behavior [8]. This difference can be further deconstructed into deterministic behavior (based on known input conditions, where the vehicle will exhibit a known behavior) and non-deterministic behavior (the exact behavior of the system cannot be determined based upon the input conditions). As it is impossible for software designers to anticipate every situation a system will one day find itself it, allowing a system to exhibit non-deterministic behavior is essential for certification. This research focus on defining a box where a system will be allowed to exhibit non-deterministic behavior.

For naval aviation, airworthiness certification authority is delegated to Naval Air Systems Command (NAVAIR) 4.0 Engineering (4.0P is the branch assigned) [9]. When a new capability (i.e., software, weapon or air frame) is acquired, and before naval personnel operate it, 4.0P must grant a flight clearance (also referred to as a SOF certification). The certification of naval aircraft follow an engineering based risk mitigation process. Aircraft subsystems, software, components and ultimately the aircraft itself are certified through an established process. Technical Area Experts (TAEs) are tasked with reviewing certification evidence (referred to as artifacts) in their individual technical areas. These reviews are rolled up into a larger flight clearance which certification officials use to certify the vehicle as a whole. When a vehicle is certified safe for flight, NAVAIR 4.0P is certifying that when given to a qualified pilot they can safely complete the desired mission of the aircraft [9].

As NAVAIR 4.0P certifies aircraft to be operated by qualified pilots, it is important to understand how the process to qualify a pilot differs from the aircraft SOF certification process. The qualification process for naval aviators (pilots) is considered to be a trust process. Unlike the civilian sector, military pilots are trusted by their Commanding Officers (COs) to complete missions critical to national interests. While each pilot is required to log a minimum amount of flight time, and show competency in aircraft procedures prior to qualification, a CO will not designate them as fully qualified until the individual has earned the trust of the CO in their decision making abilities in off nominal conditions [10].

In order for a naval autonomous aerial system to be certified to complete tasks currently reserved for qualified pilots, a new process needs to be developed that can bridge the gap between the engineering focused NAVAIR 4.0P process and the trust process currently used by COs.

Autonomy offers tremendous advantages for military aviation. But, the largest advantage will be budgetary. By eliminating the requirement to train aircrew an immediate cost savings will be achieved [11]. All military acquisition programs are governed by Department of Defense (DoD) Instruction 5000.02T and it divides the life cycle of a program into several phases. While there will most likely be a larger expenditure during the Engineering and Manufacturing Development (EMD) phase due to increased test and evaluation required, there is expected to be a dramatic savings during the Sustainment phase due to the reduced costs of Operations and Maintenance (O&M) [11]. Long term, the reduced wear and tear on aircraft from the reduced training requirement will result in aircraft spending more years in service (as their useful service life measured in flight hours). The reduced budgetary landscape, coupled with the ever increasing cost of manned platforms, has created a large appetite for autonomy in the DoD. Once an autonomous system is granted a SOF certification, it can be used for the dull, dirty and dangerous missions in place of a manned aircraft.

Anything dealing with fielding a system for the military will need to be vetted through the DoD acquisition process [11]. The process is designed to ensure that systems in the acquisition process have the proper checks and balances. In addition, current DoD regulations limit the design of autonomous weapons system without allowing the exercise of appropriate levels of human judgment over the use of force [12]. This has limited autonomous research within the DoD to systems/behaviors that comply with the regulation. The DoD and NASA use Technology Readiness Levels (TRLs) to describe maturity levels for new technologies during the acquisition process [11, 13, 14] (Table 1.1 provides a short summary of the NASA TRL levels as described by Mankins in his 1995 white paper). Prior to fielding, an autonomous system it will need to demonstrate it can perform the mission under controlled

TRL 1	Basic principles observed and reported
TRL 2	Technology concept and/or application formulated
TRL 3	Analytical and experimental critical function and/or characteristic proof-of-concept
TRL 4	Component and/or breadboard validation in laboratory environment
TRL 5	Component and/or breadboard validation in relevant environment
TRL 6	System/subsystem model or prototype demonstration in a relevant environment
TRL 7	System prototype demonstration in a space environment
TRL 8	Actual system completed and "flight qualified" through test and demonstration
TRL 9	Actual system "flight proven" through successful mission operations

Table 1.1: Technology Readiness Levels Summary Used by NASA [14]

conditions (Developmental Test (DT)), and under mission representative conditions (Operational Test (OT)). A system is considered to be at TRL 7 during DT, and TRL 8 during OT. A fielded system is considered TRL 9 [13]. However, naval flight clearance officials have not had the opportunity to evaluate an autonomous system that qualifies as TRL 8. Yet, before certification officials will approve a process that will lead to a TRL 9 system, the new process needs to be evaluated for flaws.

1.1 Motivation

The past 15 years has seen a dramatic use of aviation related automation and autonomy within academic research, yet SOF regulators have not kept pace. Autonomy has been seen as a new field where science is starting to produce results close to science fiction. The ability to research new advances via relatively low cost and easy to program platforms has spurred nearly every university to have some level of active research in this field of study. Yet, as has been seen with nearly every advance to the state of the art, academia is outpacing regulatory authorities. Despite the fact that researchers continue to develop new algorithms or autonomous capabilities regulators are reluctant to approve their use for the general public. This is despite a clear desire from the general public to increase the level of autonomy in our everyday lives.

The automobile industry is one example of the increasing use of automation in our everyday lives. Modern automobiles have several capabilities that may be considered "driver relief modes" or automation. These capabilities include, but are not limited to: Cruise control, Brake assist, and Hands-free parallel parking. While self-driving cars have been studied for decades, it was not until the Defense Advanced Research Projects Agency (DARPA) grand challenge (2005) that major advances were seen in the practical application of self-driving cars [15]. Tesla vehicles have had the hardware and software installed for truly autonomous operation since the 2016 model year. However, to operate the vehicle in autonomous mode the driver has to be at the controls ready to take over at all times for it to be legally operated [16–18].

What about aviation? Science fiction promised the general public that we would have robots flying our aircraft. But when will the general public have this opportunity? Automation has been part of aviation platforms since the beginning as the first autopilot was used for straight and level flight in 1914. A modern airliner can complete an entire flight (from takeoff to touchdown) without the pilot making any control inputs. The pilot is there for regulatory reasons and simply needs to monitor the aircraft. Academia has shown that we are now at a point where a computer can make decisions that are normally reserved for pilots. However, there currently is not an approved path defined for a SOF certification authority to certify a computer to exhibit non-deterministic behavior when it is controlling an aircraft.

The first time anything is done is always the hardest. This is especially true when asking a certification authority to accept risk. There is a large amount of perceived risk in the general public for allowing an aircraft to operate without a qualified pilot. A large body of evidence, and a solid methodology needs to be assembled for the risk to be accepted by the risk averse certification agencies. For civilian applications the FAA would be the certification authority. For naval aviation, NAVAIR 4.0P has the delegated authority for SOF certification.

1.2 Problem Formulation

Before a naval aviation acquisition (i.e., weapon, software, component, or whole aircraft) can be fielded, a SOF certification must be granted by NAVAIR 4.0P. When certification officials issue a clearance they are certifying that when the asset is used by a fully qualified pilot they can safety accomplish their assigned mission. Their process is engineering focused, and is geared to verify what the system will do under various conditions (to include the actions of the pilot/operator).

The pilot certification process is a trust process. Ultimately when a CO designates a pilot as being fully qualified they are certifying that they trust the judgment of that pilot. Naval aviators may find themselves in situations that were not anticipated, and if the aviator makes the wrong decision the ramifications may cause loss of aircraft, loss of life, or even an international incident.

Autonomous aircraft will not have a pilot in the loop. Therefore, the trust process currently inherent in certification will be lost. So how do we certify autonomy? How do we certify a system to operate without a human in the loop? Who is responsible if something unexpected happens?

In an attempt to provide valuable lessons learned to naval SOF certification officials, this research focuses on certifying an autonomous system to make decisions that are currently reserved for qualified pilots. As there currently is not a need for this type of certification, an approved method for obtaining one does not exist. Through close coordination with naval SOF officials, senior naval officers (those currently responsible for designating a pilot as being fully qualified) and the Test and Evaluation (T&E) community, we propose and exercise a methodology in the hopes that the lessons learned from this first attempt will help guide officials as they eventually develop an approved process.

1.3 Background

There have been several proposed approaches for certification of unmanned or autonomous systems. A majority of the work deals with small Unmanned Aerial Vehicles (UAVs), or theoretical methods for certifying large vehicles. One common theme is to identity errors in the software early in the design cycle since the later a defect is found the more resources (both time and money) are required to correct the issue [8, 19–22]. Many of the approaches involve M&S to determine if the software is adequate for the system requirements [20, 23–30]. Another common approach involves employing formal methods for safety critical software Verification and Validation (V&V) (e.g., run time verification [31–42] model checking [43–55] and theorem proving [44, 55–61]). Some papers have detailed methodologies for V&V for the unmanned see-and-avoid requirement, but only for a two dimensional problem [62,63]. Other proposals highlight the limitations of programming and simulating a pilot's ability to sense and accurately build their Situational Awareness (SA) during flight [64–70] then make decisions based on changing situations [33,71].

One drawback of these approaches is the limited scope of their work. As an entire approved methodology does not exist, previous work has been limited to one or two pieces of the V&V process, and most did not consult aviation certification officials. One notable exception is the work done by the Formal Methods Group at National Aeronautics and Space Administration (NASA) Langley. Currently NASA is working on, and have published, several papers on obtaining flight clearances for Unmanned Aerial Systems (UAS) to operate within the national airspace [72–74]. Their work focuses on formally defining the specification from the requirements of operation within the national airspace, and then V&V via theorem provers. This is designed to give certification officials confirmation that the software will perform per the requirements. However, their work focuses on an objective standard (such as maintain 1,000 ft separation), not a judgment task (such as interpret the environment and make the best decision). As the current SOF certification process is designed to approve a system to be utilized by a fully qualified pilot, it has been hypothesized that before a SOF certification will be granted for an autonomous system the system under test needs to demonstrate that is can perform as a qualified pilot would [75, 76]. One issue with this plan is the complexity of accomplishing it. The complexity of autonomous systems results in an inability to test under all known conditions, difficulties in objectively measuring risk, and an ever-increasing cost of rework and redesign due to errors found late in the V&V process [8].

Most of the current work to certify autonomy is based off easily definable, black and white regulations for operating in the public airspace. One example is collision avoidance, where aircraft are required to maintain a safety bubble around them to avoid other aircraft. This involves an easily definable and well documented set of requirements (such as lateral and vertical separation). These requirements do not involve pilot judgment, and can be accomplished by using data currently available via onboard systems (such as Traffic Collision Avoidance System (TCAS) and Mode C transponders).

We focus on two tasks, or missions, that are currently reserved for qualified pilots. The first is the the Confined Area Landing/Landing Zone (CAL/LZ) mission currently being executed by the USN and USMC helicopter communities. The CAL/LZ mission can be as simple as landing in an open field adjacent to a highway, or as difficult as landing between buildings in an urban setting. Prior to being certified as a Helicopter Aircraft Commander (HAC), a candidate is expected to be able to complete the mission safety. During the mission, a pilot is required to monitor multiple factors in an ever change environment and make a judgment based decision as to where to land. The second is the RETROGRADE/SCRAM task currently carried out by High Value Airborne Asset (HVAA) aircrew. A HVAA is unable to defend itself and is required to maintain a standoff range from threat aircraft. When a threat aircraft reaches a defined range, the HVAA will be required to RETROGRADE (withdraw from station in response to a threat, continue mission as able). Once the threat is no longer a factor, the vehicle can RESET to its orbit. During a RETROGRADE, the HVAA platform can continue to complete its assigned mission. When a threat aircraft reaches a defined range, the HVAA will be required to SCRAM (egress for defensive or survival reasons). A fully qualified pilot is expected to take in the information available to them (both from communications with other assets and onboard systems) to determine when an aircraft reaches one of these pre-briefed limits. This decision can be considered a judgment decision based on the fidelity of the information available.

To complete a judgment task, a pilot needs to be able to interpret the available information in flight and build a mental model of their environment. This mental model is called Situational Awareness (SA). An understanding of SA as it relates to aviation is critical to understanding how it will relate to the certification of autonomy. One of the most commonly accepted definitions for SA is "the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future [77]". During flight school, student naval aviators are taught that SA in aviation is being able to accurately diagnose what is happening around them and predict what will happen in the immediate future, thus enabling them to perform the assigned mission safely. Students with high SA are able to "stay ahead of the aircraft", while students with low SA tend to seem to be "holding onto the stab" during flight. From their first flight, aviators learn to use every available resource to develop their SA (e.g., radio calls, aircraft instruments, visually scanning outside of the aircraft, onboard radar, Electro Optical/Infrared (EO/IR) sensors and seat of the pants feelings). Prior to obtaining full qualification, a naval aviator will have proven to their CO that they can develop their SA to an appropriate level that they can safely complete their assigned mission during off nominal conditions [10]. The measurement of SA has proven to be an intangible, and largely subjective concept. Pilots quickly learn that the only way to know exactly the level of their current SA is when they realize that they have none. When a pilot's SA is high (i.e., they have an accurate understanding of the environment they are operating in) they can make sound aeronautical decisions. However, when a pilot's SA is low (which they may or may not know at the time) their aeronautical decisions may not be sound. Autonomous vehicles will use their sensors to build SA of their environment. When properly designed sensors are operating at 100% the contributions they provide to the vehicle SA should be adequate to support sound aeronautical decisions. However, at some point of sensor degradation the vehicle SA will no longer match reality.

1.4 Certification Methodology

This dissertation was prepared in close coordination with naval SOF clearance officials to determine a path forward for certifying autonomy in naval aircraft. A method for certifying a vehicle to make decisions when a qualified pilot/operator is not in the loop does not currently exist. We proposed, and certification officials concurred, the following methodology as a possible avenue for certifying autonomy in the hopes that lessons learned from its exercise will help develop an approved process before the first autonomous system is acquired by the Navy:

1. Define the requirements (normally reserved for a pilot) to execute autonomous behavior. These requirements must be developed through coordination with SOF certification officials, the naval T&E community, and fleet officials who currently certify pilots as fully qualified. A specification will then be developed that can be used to verify the requirements have been completely and accurately specified.

2. Develop the clearance envelope where the system will be allowed to exhibit non-deterministic behavior (the exact behavior of the system cannot be determined based upon the input conditions). If the system were to encounter the edge of this envelope it would revert to deterministic behavior (based on known input conditions, the vehicle will exhibit a known behavior).

3. Analyze the specification to ensure the requirements of the system are met.

4. Develop a protocol/set of control laws with traceability to the verified specification. This way formal methods will satisfy the requirements of the system, as the protocol/control laws will have formally verified properties.

5. Limited M&S of the algorithms/control laws as a risk reduction tool prior to flight test. This will attempt to show the system will display non-deterministic behavior only while it is within the clearance envelope.

6. Design the process for flight test. Most conventional flight test techniques

are designed for a pilot to test an unproven system. In this case, test points will need to be developed that demonstrate under controlled (DT) and operationally relevant conditions (OT) the system under test can complete the assigned mission.

7. Execute DT and OT on the autonomous system under test.

8. Full report of the tests conducted on the system under test.

1.5 Contributions

As I am currently a senior naval officer with contacts and established relationships within the naval acquisitions community (to include T&E Community Leadership, SOF certification officials, and the NAWCAD center for autonomy) this research was given many unique opportunities not normally afforded to University studies. These opportunities included access to senior officials for interviews and guidance, access to existing data sets within the Navy, and access to DoD approved M&S environments.

The original contribution to knowledge contained in this dissertation include:

- Proposed methodology for obtaining a naval aviation SOF certification allowing a decision engine to complete a task currently reserved for a qualified pilot. Then the exercise of the methodology to help build a path forward for certifying autonomy. Currently the United States Navy (USN) does not have a path forward for certifying autonomy. This contribution will influence future certification standards and procedures for this emerging requirement.

- Definition of the requirements a decision engine must complete if it were to be approved to complete the CAL/LZ mission autonomously (a task currently reserved for a qualified pilot). Then use of a formal methods approach to ensure the actions taken by a developed protocol will satisfy the requirements defined. This contribution exercised the first four steps of the methodology proposed in Section 1.4 and provide artifacts to certification officials for a possible SOF clearance allowing a decision engine to complete a mission currently reserved for a qualified pilot.

- Development of flight test matrices (one for DT and one for OT) for an autonomous vehicle to complete the CAL/LZ mission. Followed by analysis of both DT and OT flight test data of an autonomous vehicle completing a task currently reserved for a qualified pilot (CAL/LZ mission). This contribution exercised the last three steps of the methodology proposed in Section 1.4 and provided artifacts to certification officials for a possible SOF clearance allowing a decision engine to complete a mission currently reserved for a qualified pilot.

- Development of an objective measure for autonomous vehicle SA that accounted for sensor degradation within a Department of Defense (DoD) recognized M&S environment. The measure specifically evaluated the effects of sensor degradation on error distance of a fused track of a threat aircraft. We used Design of Experiments (DOE) to determine the effects of sensor degradation and produce a set of predictive equations for the error distance of the fused track. Then used Subject Matter Expert (SME) opinion to define the point at which (within this scenario) the fused error distance was inadequate to make a decision currently reserved for a qualified pilot. This contribution exercised the fifth step of the methodology proposed in Section 1.4 and provided results that if confirmed during flight test could have lead to a SOF clearance allowing a decision engine to complete a mission currently reserved for a qualified pilot.

1.6 Dissertation Outline

This dissertation is structured as follows. Chapter 1 has been an introduction to the research. Chapter 2 is a literature review focused on developing a path forward for certifying a decision engine to complete the CAL/LZ mission in a large rotorcraft. Chapter 3 details the process of certification from requirements development to the establishment of a protocol for an autonomous air vehicle to complete the CAL/LZ mission and will exercise the first four steps of the methodology proposed in Section 1.4. Chapter 4 presents flight test data, and the analysis of that data, of an autonomous air vehicle completing the CAL/LZ mission (both developmental and operational flight test data), and exercises the last three steps of methodology proposed in Section 1.4. Chapter 5 details the development of an objective measure for determining adequate SA of an autonomous air vehicle to complete a task currently reserved for qualified pilots (identifying the range for RETROGRADE or SCRAM) in an M&S environment, and exercises the fifth step of the methodology proposed in Section 1.4. Chapter 6 summarizes the work, provided a list of original contributions and provides an outlook for future work related to this topic.

Chapter 2: Literature Review

2.1 Overview

Certifying an autonomous controller to complete tasks currently reserved for qualified pilots is on the critical path for autonomous aerial vehicles to be fielded. Since the dawn of aviation, many of the innovations we currently take for granted came from the military (some examples include: radar [78], medevac air ambulance [79], jet engines [80], glow sticks [81], and advanced night vision technology [82]). For this reason, and due to classification issues, we initially focused our research on defining a path forward for an autonomous controller to accomplish a task currently carried out by the USN and USMC helicopter communities: Landing a full size helicopter autonomously in an unprepared LZ. This literature review is focused on defining the issues associated with accomplishing that research.

This chapter is structured as follows. Section 2.2 is a brief review of automation within tactical naval aviation. Section 2.3 covers the naval certification process (both aircraft and aviator). Section 2.4 deals with building trust in autonomous systems leading to certification. Section 2.5 discusses bringing autonomy to military aviation. Section 2.6 delves into formal methods research. Section 2.7 is a broad overview of other topics for autonomy in military aviation. Section 2.8 covers the helicopter landing mission. Finally, section 2.9 discusses the HAC qualification process in detail.

2.2 Review of Automation within Tactical Naval Aviation

As previously noted, automation has been part of aviation since its early days. Pilot relief modes such, as an autopilot, are forms of automation. Unlike helicopters, most naval tactical aircraft are single piloted with only a single seat. However, some will have an additional seat for a Naval Flight Officer (NFO). The NFO can assist in managing aircraft systems but does not have access to flight controls. This single piloted nature, and the ever increasing workload pilots face, has manifested a need for increased automated functionality within the aircraft.

As automation is designed to make it easier for a human pilot to complete tasks, it is only natural that the high levels of automation are installed in carrier based tactical aircraft. Examples include autopilot/pilot relief modes within the F/A-18 family of aircraft, the automated carrier landing functionality inherent to the precision approach and landing System, and the X-47 demonstration program.

2.2.1 Autopilot/Pilot Relief Modes in the F/A-18 Hornet Family of Aircraft

The F/A-18 Hornet family of aircraft include the F/A-18 A-D (Legacy Hornet), F/A-18 E/F (Supper Hornet) and EA-18G (Growler). The Legacy Hornet began development in the 1970s. It was designed to be a multi role single seat aircraft. After the lessons learned in Vietnam developers made numerous provisions to make the aircraft easier to fly. This gave the pilot more capacity to manage the various aircraft systems. These provisions are referred to as pilot relief modes. As of 2020, the Super Hornet and Growler are still being produced with pilot relief modes in mind. Figure 2.1 is an image of a two place EA-1G Growler during transonic flight test in 2009.

Figure 2.1: EA-18G Growler During Flight Test in 2009 [83]

Pilot relief modes include, but are not limited to: mach hold, calibrated airspeed hold, altitude hold, heading hold, flight path hold, and flight plan coupling. During an airways navigation flight from St. Louis to Phoenix, with limited preflight planning and through the use of the pilot relief modes, a pilot would only need to make limited flight control inputs. During combat, the judicious use of pilot relief modes enables the pilot to spend more of their time focused on the mission and less actually flying the aircraft.

While the Legacy Hornet has limited direct connections to the flight controls, the Super Hornet and Growler are completely fly-by-wire. In a fly-by-wire aircraft all flight control inputs are transmitted to the flight control computer which ultimately makes the decision on what flight control surfaces to actuate. All members of the Hornet family are extremely software dependent. Each new software patch, or update, requires rigorous regression testing. This testing is done through M&S, in various laboratories with hardware in the loop, and ultimately through flight test. As the V&V community begins looking at methods for V&V of autonomous systems there needs to be a new way of certifying these systems. While some testing will still be completed via current procedures, new methods will need to be developed and employed.

2.2.2 Automated Landing of Tactical Jets on Aircraft Carriers

Automated landing is not a new concept for naval aviation. All USN CVNs, LHDs, and LHAs are equipped with the Precision Approach Landing System (PALS) (also referred to as the Automatic Carrier Landing System (ACLS)). PALS enables aircraft to "couple" with the ship and allow a qualified pilot to automatically (with no control inputs) land their aircraft. During an ACLS approach the pilot has the responsibility to guard the controls and take control if there is a perceived or actual malfunction. For CVN aviation, pilots are not allowed to use the this mode (except in extremes) unless they are an experienced fleet aviator having completed at least one six month deployment aboard ship. This is the risk mitigation step leadership put in place for safety. A new pilot may not know when an approach is unsafe, and therefore not take control from the automated system during a malfunction.

Understanding the certification of the PALS capability may give insight on future automation or autonomy certification. Unlike software and hardware certification on aircraft (which only require certification on initial development or modification), the PALS system is required to be re-certified every 24 months on a CVN and 46 months on a LHA or LHD [84]. A PALS certification effort demonstrates, through PALS certification tests, that the system can assist pilots of qualified aircraft to accomplish safe manual and automatic approaches (if capable) to touchdown above the established weather minimums and within the determined operational envelope [84].

A full certification of a CVN PALS may be required for several reasons. Some of them include but are not limited to: a new aircraft to be controlled (such as the F-35C Joint Strike Fighter), PALS equipment move/upgrade, or following ship overhaul. Full certification is an extensive proposition, requiring multiple weeks in port and over a week at sea for the V&V process [85].

The Carrier Suitability Department at VX-23, in conjunction with the Naval Air Traffic Management Systems Program Office (PMA-213) manage the certification of all PALS onboard ship or at equipped Naval Air Stations. Full certification of a PALS onboard a CVN is administered via a test plan. While each certification may be covered by an individual test plan, they are all similar in level of effort to the master PALS test plan. As of December 2017, the governing test plan for PALS certification was 127 pages. It lists the attributes being examined as well as how the various tests will be conducted. When any part of the certification plan changes (such as equipment, supporting flight clearances or test personnel) an amendment is required prior to actual flight test. This document is designed to standardize the evaluation and is used as a risk mitigation tool for certification officials [86]. For a full certification, 30-40 flight hours are anticipated. A one page test matrix (Figure 2.2) is further detailed by a 48 page Detailed Method of Test (DMOT).

Task	Test #	Sub-Task	Objective	Cert. (1)	Ver. (1)	Ext. (1)	ITPs	Pier-Side (2)	At-Sea (2)	Risk Category
1.0 Aircraft Checkout	1.01	Pre-Flight Ground Check	Verify A/C ACLS system functionality prior to takeoff.	N/A				N/A		A
	1.02	ATC Check	Verify ATC functionality.							
	1.03	ICLS Check	Verify A/C ICLS functionality.							
	1.04	Mode II Buildup	Buildup to Mode I approaches to deck, evaluate ability to track A/C BCN.							
	1.05	Mode IA Buildup	Buildup to Mode I approaches to deck.							
	1.06	Mode I Check	Verify A/C ACLS functionality.							
2.0 CVN Cert. Tests	2.01	IFLOLS H/E	Verify IFLOLS H/E values.	✓	✓	✓	✓	N/A		N/A
	2.02	IFLOLS Basic Angle	Verify IFLOLS 3.5 & 4.0 deg basic angle.	✓	✓	✓	✓			
	2.03	AN/SPN-46 Inertial Alignment	Align AN/SPN-46 stabilized coordinate system to Earth's surface.	✓	✓	✓	✓	(3)	(3)	A (Shipboard) B (Pierside)
	2.04	AN/SPN-41 Alignment	Align AN/SPN-41 azimuth & elevation to ACLS.	✓	✓	✓	✓	✓	✓	
	2.05	Mode I Control Performance	Evaluate Mode I control of A/C.	✓	✓	✓	-	✓	✓	
	2.06	TDP Performance	Evaluate Mode I TDP performance & completion rate.	✓	✓	(4)	-	-	✓	
	2.07	Stabilization Source Evaluation	Evaluate AN/SPN-46, AN/SPN-41, and IFLOLS performance on all stabilization sources.	✓	✓	-	-	-	✓	
	2.08	AN/SPN-46 Tracking Mode Evaluation	Evaluate AN/SPN-46 tracking modes.	✓	✓	-	-	✓	✓	
	2.09	DAIR Handoff Evaluation	Verify functionality of AN/TPX-42 DAIR Handoff.	✓	✓	-	-	✓	✓	
	2.10	DAIR Auto-Acquisition Evaluation	Verify functionality of AN/TPX-42 DAIR Auto-Acquisition.	✓	✓	-	-	✓	✓	
	2.11	WOD Envelope Establishment/ Expansion	Establish WOD envelope (certification), expand WOD envelope (verification).	✓	(5)	-	-	-	✓	
	2.12	AN/SPN-41Elevation Coverage	Verify AN/SPN-41 elevation coverage	(6)	-	-	✓	✓	✓	
	2.13	AN/SPN-41 Azimuth Coverage	Verify AN/SPN-41 azimuth coverage.	(6)	-	-	✓	-	✓	

Notes:
(1) A ✓ indicates the test is *required* to be completed for a certification, verification, or extension effort.
(2) Unless otherwise noted, a ✓ indicates the test *may* be conducted pierside or at-sea.
(3) AN/SPN-46 Inertial Alignment is *required* to be completed at-sea to issue or extend a certification. The test point *may* additionally be conducted pierside to obtain an initial indication of adjustments that may be required during at-sea testing.
(4) TDP Performance is *required* for an extension effort only if adjustments are made IAW Table 4~~Table 4~~.
(5) WOD Envelope Expansion may be conducted during a verification effort to expand a limited WOD envelope established during a previous certification.
(6) AN/SPN-41 Elevation and Azimuth coverage will be conducted for new AN/SPN-41B installs or IAW tasking in Reference (1).

Figure 2.2: Test Matrix for CVN PALS Certification [86]

The DMOT details the particulars of the test conditions and gives evaluation metrics for certification. While several metrics are evaluated, the hook touchdown point is the main test metric examined for PALS certification. A Nimitz class carrier has four arresting wires (number 1-4 from aft of the ship to forward) spaced about 50 ft apart. The ideal hook touchdown point is halfway between the 2 and the 3 wire. See Figure 2.3 for a pictorial of the ideal touchdown point.

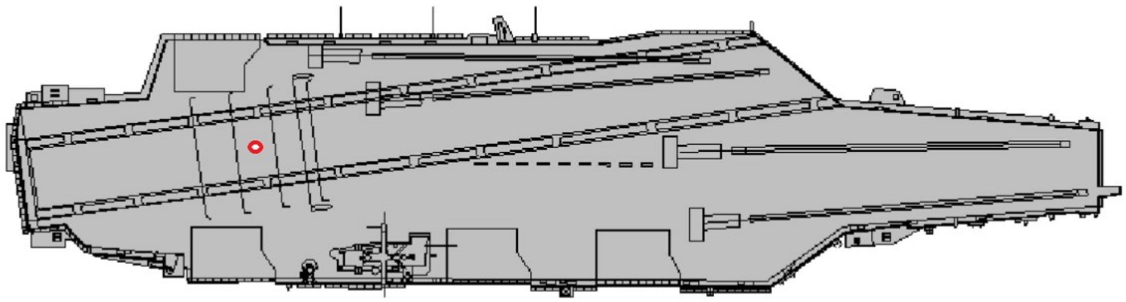

Figure 2.3: Graphical Depiction of the Ideal Hook Touch Down Point for PALS Approach on a Nimitz Class Aircraft Carrier [87]

When an aircraft approaches the flight deck, it is required to maintain constant angle of attack (AOA). This AOA is designed so that the arresting hook and the main landing gear touch down at the same time. In addition, proper AOA puts the hook at an optimum angle to properly engage the arresting wire. To accomplish this the systems onboard the CVN must be set for the actual aircraft type on approach. This delta between the "Beacon Flight Path" and the "Hook Flight Path", as depicted for an F/A-18 in Figure 2.4, will not be the same for all types of aircraft using the system and the ACLS must be matched to the aircraft on approach.

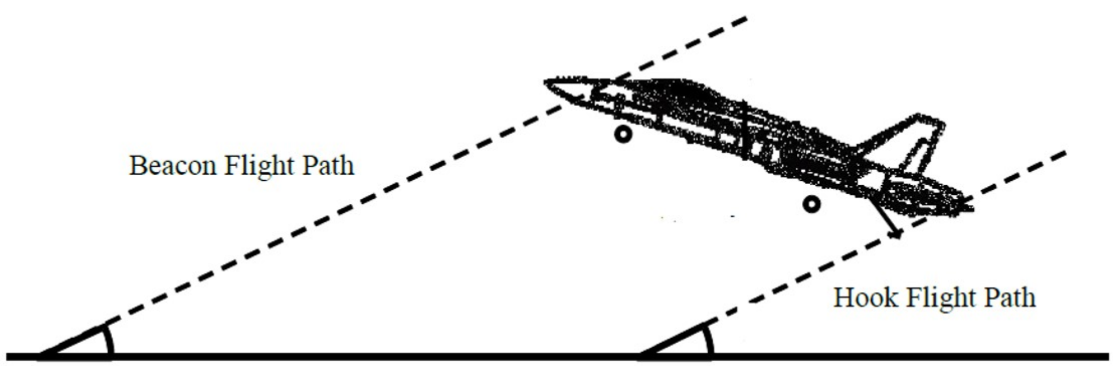

Figure 2.4: Graphical Depiction of the Difference Between the Beacon and Hook Flight Path on PALS Approach [86]

The objective of PALS is to put the arresting hook of an aircraft in a position to safely engage one of the arresting wires onboard the aircraft carrier. The main certification metric is the achievement of a 95% confidence interval that the mean hook touchdown point is within 15 ft of the desired touch down point [86]. Figure 2.5 is included for reference to an established metric used by naval certification officials for automation onboard CVNs. Test engineers use it for evaluation of the performance of the system under test.

The discussion of how Naval certification officials V&V PALS onboard aircraft carriers is included as an example of how the use of a metric for automation certification. Provided the system is set up properly (uses established norms for equipment and environmental conditions), and the CVN passes its bi-annual certification, pilots can use the system to land their aircraft onboard a CVN with no pilot input. As the military begins grappling with ways to certify autonomy this may serve as a possible path towards certification.

2.2.3 X-47 CVN Demonstrator

Obi-Wan Kenobi was a wise Jedi. He had numerous quotes, but the one that is most relevant to this body or research was "Flying is for Droids" [88]. The X-47 demonstrator (Figure 2.6) program had a limited goal. Demonstrate that a UAV could operate in the CVN environment, to include arrested landing and catapult launches. Unfortunately a majority of the technical achievements of the program are protected under propriety agreements with its designers. Through several interviews

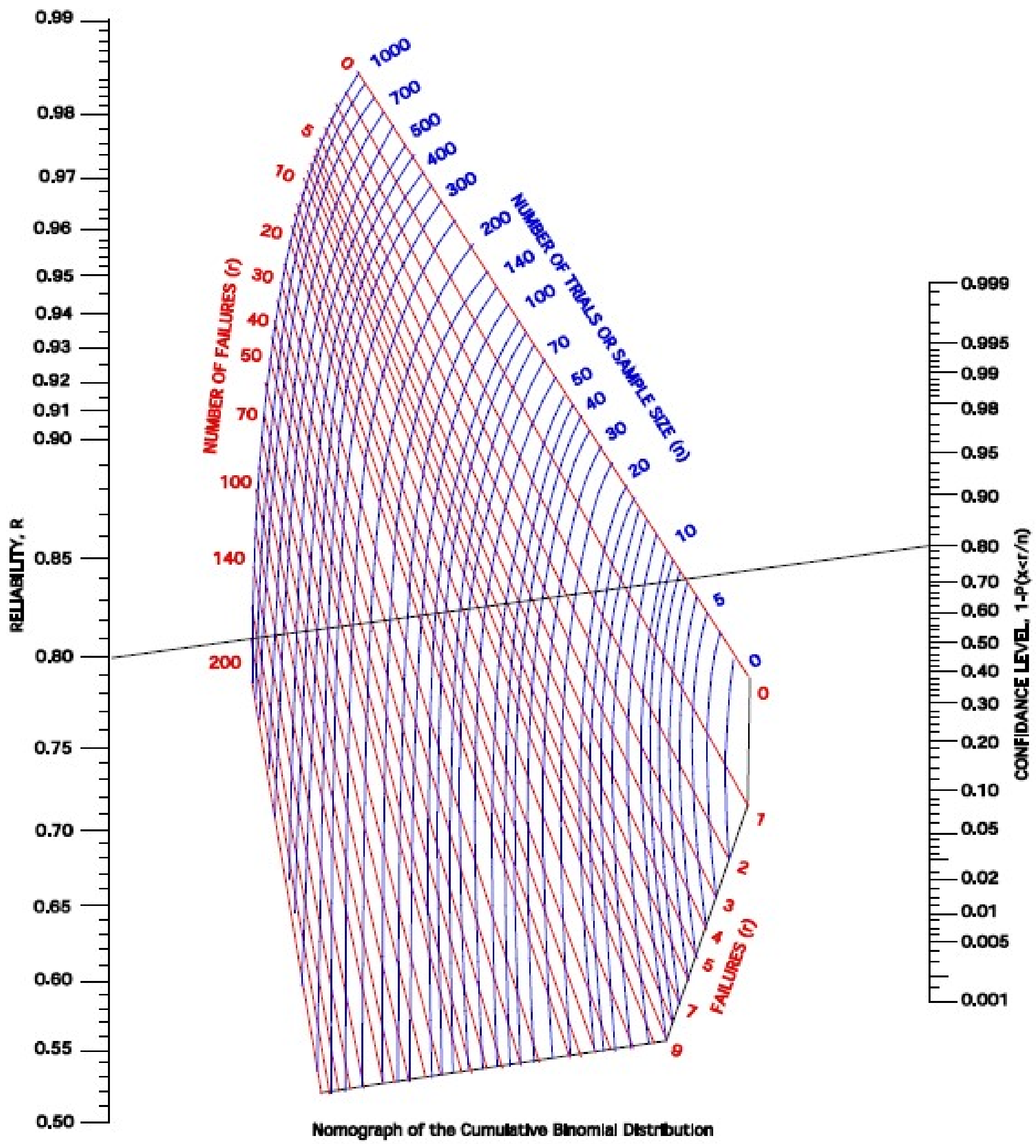

Figure 2.5: Extract From the DMOT Portion of the PALS Certification Test Plan Detailing the Requirements [86]

with members of the flight test team, we were able to construct the issues related to certifying the X-47 for flight in the national airways systems and operations around a CVN.

The X-47 was designed with limited utility. It had a simple mission: Demonstrate that an unmanned aircraft could operate in the carrier environment. To

Figure 2.6: First Landing of a Large Fixed Wing Drone on Board an Aircraft Carrier, 10 July 2013 [89]

accomplish this mission, the test team used simplicity for airworthiness certification. They programmed the X-47 to exhibit deterministic behavior. All emergency procedures (for hardware or software failures) were hard coded into the system. A human was always in the loop, or able to override the actions of the onboard computer system. In cases where the system lost link with its controllers it would either perform a preplanned, and airspace cleared, approached back to one of three select airports or it would ditch in the ocean.

In the end, the deterministic behavior exhibited by the X-47 was able to achieve limited certification by the Navy and the FAA to accomplish the demonstration tasks for which it was developed. As part of their ongoing research and IRAD investment, Northrop Grumman (prime contractor for the X-47 demonstrator) have continued work on the problem of launching and recovering a large UAV from a CVN. As part of their IRAD investment they have started working on the V&V problem. They have proposed breaking the problem down to a number of steps, V&V the subcomponents of the model, then V&V the entire model. Their approach may work, however it faces the same questions and concerns as other currently proposed V&V techniques for autonomous systems [46].

For the purposes of this research, deterministic behavior shall be defined as "based on a known input condition, the vehicle shall perform a known behavior." This can be thought of as basic "if, then, else behavior" and to some extent all unmanned vehicles will be programmed with this behavior. However, how to we push the outside of this envelope? In this research, we examine how to certify a system that exhibits limited non-deterministic behavior. Test assets will be given an envelope in which it can make its own decisions. Determining how certification officials will accept the risk associated with this is on the critical path for certifying autonomy.

2.3 Naval Certification Processes

2.3.1 Naval Aviation Aircraft Certification Process

Currently the US military uses the T&E community for V&V of new capabilities. The military branch which will eventually field the capability are tasked the airworthiness certification for the new capability. As this research focuses on naval aviation, this discussion will focus on the practices and policies which govern naval aircraft. NAVAIR 4.0P, located on NAS Patuxent River, is the ultimate flight clearance authority for naval aircraft and naval aviation systems. The governing instruction of naval aircraft certification is the NAVAIR Airworthiness and Cybersafe Process Manual (NAVAIR Manual M-13034-1) [9].

There are two types of flight test currently employed when a capability is added to a naval aircraft, Developmental Test (DT) and Operational Test (OT). DT is performed by trained test pilots. A trained military test pilot is defined as a graduate of the USNTPS (NAS Patuxent River, Maryland), USAFTPS (Edwards AFB, California), Empire Test Pilot School (Boscombe Down, England), or Ecole de Personnel Navigant D'Essais et de Reception (EPNER, located in Estres France). TPS teaches future test pilots classical test techniques to evaluate experimental aircraft and new capacities to already fielded aircraft.

DT test points are normally controlled, and designed to determine if the capability meets individual specifications and requirements. An example of a developmental test requirement might be: "The aircraft will achieve a speed of 1.4 Mach at 10,000 ft MSL during a level acceleration." This requirement has a clear condition (1.4 Mach at 10,000 ft MSL), and a clear method to achieve it (level acceleration). Once a new capability (full aircraft, new software load, or weapon) has successfully demonstrated that it meets the required DT specifications it may transition to OT.

The purpose of OT is to ensure that the new capability is suitable for the mission it is expected to complete. An OT pilot is a fleet experienced aviator. They typically have been in the service for approximately seven years, and have complete a fleet tour (deployable squadron). For a new capability to be deemed suitable (and pass OT) it must be able to perform the mission in a fleet representative environment. An example of an OT requirement may include: "The aircraft must be able to integrate into a multi-plane strike verses a remote target in a contested environment." Modern OT differs from DT in several ways beyond simply the training required for its aircrew. DT is designed to ensure the capability matches the requirements levied by the government. OT is designed to ensure that the military can use the capability to complete its mission. It is possible for a capability to successfully pass DT, but fail OT. This is one of the reasons the DoD only requires OT.

Following T&E, and the successfully demonstration of the capability, NAVAIR 4.0P will issue an airworthiness certification for the new capability. However, this paradigm is based on a number of assumptions. NAVAIR issues the certification for the hardware and software onboard the aircraft. This certification assumes that there is a qualified human (either pilot for manned aircraft or controller for UASs) making the actual decisions on how the system will be employed.

2.3.2 Naval Aviator Certification Process

The certification of aviators (pilots in the USN) is the responsibility of the Chief of Naval Air Training (CNATRA) (commanded by a one or two star admiral) and that of Commander Naval Air Forces (CNAF) (commanded by a three star admiral). To be fully qualified, a pilot must pass a number of preliminary certification standards before they are able to formally take the safety of flight responsibilities and sign as the Pilot in Command (PIC) of a naval aircraft.

While the PIC of a single seat tactical jet can sign for their aircraft fairly early in training, it is not until they are designated as a section (two aircraft operating together) leader that they are considered to be fully qualified. Typically a new section leader will have completed a CNAF approved tactical flight syllabus, log a prerequisite number of flight hours in model, and earned their CO's trust in their judgment during unpredictable situations. It is standard for a pilot to achieve this qualification by two and a half years in their fleet squadron. Failure to achieve this qualification in the specified time may result in a board of review on the aviators flight status.

Unlike tactical jet aircraft, most military helicopters typically have two seats. A helicopter pilot is considered fully qualified once they are designated a Helicopter Aircraft Commander (HAC). Prior to being designated as a HAC, the pilot is required to successfully complete numerous schools and will have demonstrated to numerous evaluators they are proficient. Like tactical jet pilots, they must complete a number of prerequisites demonstrating their skills in the cockpit. However, unlike tactical jet pilots, to be fully qualified they must pass a "HAC Board" which is led by the squadron leadership to test their judgment under pressure in a variety of situations.

Regardless of the track an aviator takes to become fully qualified, the outcome is the same. They must complete an extensive syllabus that convinces leadership to trust their judgment. Once a naval aviator is considered fully qualified, they are trusted to make decisions based on their past experiences. The CO has trust in the individuals ability to build their SA during off nominal conditions, and make sound aeronautical decisions based off that SA. It is only at this time that leadership (typically a squadron CO) will designate them as fully qualified.

How can these process be transitioned for certifying autonomy? How can certification officials be expected to allow a system that does not possess a learning capability to make decisions that a pilot currently make? If it does have some level of machine learning ability, how can certification officials continue to trust that it was perform within given parameters once the software evolves?

2.3.3 NAVAIR 4.0P Certification Process

Currently, when an aircraft is certified safe for flight (when operated within established limits, it will not break down or cause a danger to the general public) it is assumed that they will be operated by a qualified pilot (or operator in the case of large UAVs such as Global Hawk or Predator). Academia, and now industry, have developed software and hardware solutions that are capable of making decisions based on information provided by its sensors (such as where to land) and exhibit non-deterministic behavior.

How do we certify a decision engine (the computer acting as the PIC) to safely accomplish tasks currently reserved for a qualified pilot? For a large autonomous rotorcraft these tasks will include takeoff, path planning, obstacle avoidance and landing spot selection. While academia, industry, and the military have proposed advances in the field, and paths forward, policy makers have not acted. A majority of this work relies heavily on modeling and simulation in the V&V process [8, 23, 25, 90–92]. Most of the presented methods did not directly involve the certification authority in their work. Military certification officials are normally considered risk adverse. Unless the method for certification gives the official enough justification that adequate risk mitigation has been taken, they are reluctant to certify it safe for flight. Most of the methods proposed are M&S based. While M&S provides insight into the performance of a system, M&S alone will not mitigate the risk for certification officials.

However, the main reason that certification officials have not acted is because a truly mature autonomous system has not advanced to a point where it would require certification. Therefore, there has not been added pressure for officials to accept the risk. As we have seen in the past, the civilian certification officials (FAA) will not act until there is an overwhelming demand from the general public (this was seen when they shifted the pilot retirement age from 60 to 65 for pilots). The steps will need to be small, with limited scope. But once enough of them are taken, we should be able to certify a system of sensors and software to accomplish tasks (and eventually entire missions) currently reserved for a qualified pilot.

Currently the FAA is regulating UAS (other than model aircraft) via Part 107. For certification to operate a UAS must have a qualified operator within visual line of sight (an onboard camera cannot satisfy the see-and-avoid). At all times the small unmanned aircraft must remain close enough for the remote operator to see it unaided by any device other than corrective lenses. Part 107 only deals with UASs under 55 pounds. As of January 2020 there was not a FAA regulation for larger

UASs. In addition to a number of other stipulations in the regulation, there does not exist a path for certification for a UAS that does not have a remote "pilot in command" [1].

For naval aviation, airworthiness certification authority is delegated to NAVAIR 4.0 Engineering (4.0P is the branch assigned). When a new aviation related capability (software, weapon or airframe) is acquired, and before naval personnel operate it, 4.0P must grant a flight clearance (also referred to as a safety of flight certification). They have established processes where Technical Area Experts (TAEs), who have been given the authority in their subject fields, review relevant artifacts prior to approving their portion of a flight clearance. Artifacts can be something as simple as a SME opinion, or as complicated as detailed engineering analysis. Often an artifact is a large body of data characterizing the performance of a system. In the end, artifacts exist to quantify the system and allow the certification official to determine the risk they will be accepting.

Following initial conversations with NAVAIR 4.0P leadership, we began discussions with naval airworthiness authorities for possible avenues of granting a flight clearance for an autonomous controller (the decision engine) of a large rotorcraft (H-1 or similar sized helicopter) to complete a task that is currently reserved for fully qualified pilots. For a large portion of this dissertation, the flight clearance will be focus on accomplishing a mission relevant task: Landing in an CAL/LZ. Initial discussions developed the following list of TAEs that would be required for this limited mission focused flight clearance:

- System Safety
- Software Engineering
- Flying Qualities
- Flight Controls
- Avionics Systems Engineering
- Core Avionics
- Human Systems Interaction
- Class Desk

This research will add to the body of knowledge for how to develop artifacts the various TAEs will require prior to issuing a flight clearance for a naval aircraft to operate autonomously. There are several different types of artifacts that TAEs may use. In this research, various methods of artifact development will be used in an attempt to determine a viable method for autonomous certification. Prior to accepting the risk associated with this first of its kind flight clearance, the TAEs will require a large number of artifacts to mitigate identified risk areas. Ultimately each TAE will have to sign off on their SME area prior to a safety of flight certification.

For the respective TAEs to certify their subject area, several challenges will need to be overcome. In the words of the former chief engineer of the USAF, "It is possible to develop systems having high levels of autonomy, but it is the lack of suitable V&V methods that prevents all but relatively low levels of autonomy from being certified for use." [93] The AFRL funded a study asking a question regarding the state of possible processes for certification of UASs which employ machine learning or autonomous functionality through some sort of evidence based licensure process. These categories were [94]:

- Formal Methods
- Requirements and Metrics
- Normative Oracle Generation
- CoActive Design
- Implications of Learning Autonomous Systems
- M&S Considerations for Licensure of Autonomous Systems

All or some of these categories will be required for the individual TAEs to accept the risk associated with certifying the autonomous functionality described.

Further work has been done in this research area by The Autonomy Community of Interest Test and Evaluation, Verification and Validation (TEVV) Working Group. The TEVV working group was made up a collation of the willing. Membership included all of the US armed services major research facilities and T&E organizations. In 2015 this working group published vital definitions, challenges, and gaps for V&V of autonomous systems [8]. The challenges associated with V&V of autonomy included [8]:

- State-Space Explosion

item Unpredictable Environments

- Emergent Behavior
- Human-Machine Communication

The gaps identified by the TEVV working group included:

- Lack of Verifiable Autonomous System Requirements
- Lack of Modeling, Design, and Interface Standards
- Lack of Autonomy T&E Capabilities
- Lack of Human Operator Reliance to Compensate for Brittleness
- Lack of Run Time V&V During Deployed Autonomy .

2.4 How to Build Trust in Autonomous Systems Leading to Certification

Trust is vital for certification. When a military commander certifies a pilot as fully qualified they are bestowing their trust on that pilot. Following qualification, the pilot is expected to use his judgment to make decisions based on their experiences. When dealing with autonomy, trust is not inherent and certification is not business as usual. For commanders to trust that an autonomous system will perform as a pilot would will require methods and metrics different than what they are accustomed to.

The Defense Science Board identified trust as an integral requirement for the use of autonomous systems by the DoD. “The decision for DoD to deploy autonomous system must be based both on trust that they will perform effectively in their intended use and that such use will not result in high-regret, unintended consequences. Without such trust, autonomous systems will not be adopted except in extreme cases such as mission that cannot otherwise be performed. Further, inappropriate calibration of trust assessments – whether over-trust or under-trust during design, development, or operations will lead to misapplication of these systems. It is therefore important for DoD to focus on critical trust issues and the assurance of appropriate levels of trust [91].”

Autonomy is a new concept for the DoD. When employing military systems in the field, someone is ultimately responsible for the actions of that system. This may include the actual military member employing the technology or the individual that certified it for use. Having a system that exhibits non-deterministic behavior inherent in autonomy is a new concept for certification officials. Trust needs to be built prior to the use of autonomy within the DoD.

The robots from the classic *Jetsons* cartoon and the machines that ran civilization in the science fiction classic *Metropolis* are examples of how science fiction has influenced the general public as to the capability of autonomous systems. While we may not have a robotic maid, we now have vacuum cleaners like the Roomba that automatically clean our floors. The American public is in love with our automobiles. Nearly every family has at least one car. Science fiction promised us self-driving cars, but as of October 2020 a truly autonomous car was not certified for operation on our nation's roads.

While all Tesla models since 2016 possess the hardware and software required to operate in *Autopilot Mode*, they are not certified to operate without a qualified driver at the wheel. The driver is required to be ready to take over if the system puts the vehicle in a dangerous situation. This method lead to millions of miles of casualty free diving. However, on 7 May 2016, a 2015 Tesla Model S was involved in a fatal collision in Florida. The vehicle was operating in *Autopilot Mode*, and the automatic emergency braking system did not provide any warning or breaking, and the Tesla driver did not apply any braking or control inputs. The incident was an example of a human putting too much trust in the functionality of their vehicle. In the National Highway Traffic Safety Administration report, the DoT did not find any fault with the vehicle. When operating in *Autopilot Mode* the driver must be ready to take over for the vehicle at any time. While these vehicles use autonomy (the vehicle senses its environment and makes decisions for how it will proceed) the manufacturer, and certification officials, put the ultimate responsibility of the vehicles actions on the driver not the vehicle itself [16].

How do we built trust in autonomy which can lead to eventual certification? The automotive world has a simple plan (despite recent mishaps): Demonstrate how self-diving cars can perform just as safely as human drivers. This method is still ongoing. But is there a better way, a way that may be used for military systems?

In 2015, researchers from the University of Illinois at Urbana-Champaign published a paper in Human Factors titled *Trust in Automation: Integrating Empirical Evidence on Factors that Influence Trust*. The paper itself was a survey of 101 separate papers which involved humans working with automated systems to achieve goals with some level of trust. In it, Hoff and Bashir [95] developed a three-layered framework for conceptualizing trust variability (Figure 2.7). The three layers were dispositional, situational, and learned. These three layers can be used when referencing the use of automation and autonomy in naval aviation certification.

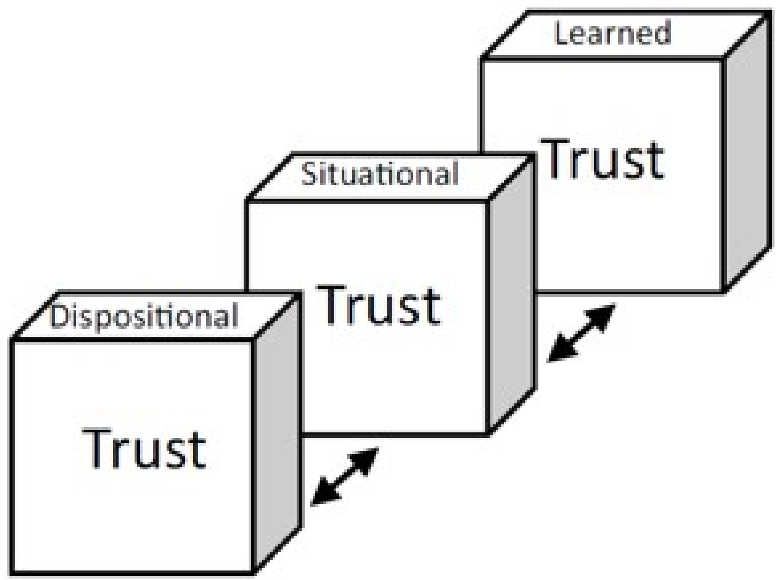

Figure 2.7: Three-Layered Framework for Conceptualizing Trust Variability as Presented in Reference [95]

Dispositional trust is a relatively stable quantity which is influenced by culture, age, gender, and personality traits. When it comes to trusting autonomy in naval aviation it can be assumed that the amount is dispositional trust certification officials may have depends on the influencing factors. It can be assumed that current certification officials, who are not accustomed to autonomy and are nearing the end of their careers, will have less dispositional trust in autonomy than the academic research community based on their experiences as a certification official and within their day to day lives.

Within their research, situational trust dealt with both the benefits and risks of using automation depending on the situation. Naval aviation has already accepted situational trust for automation. An example of this is the PALS program discussed in references [2, 86]. If the pilot was having difficulty landing an aircraft, leadership can direct them to couple their aircraft with the CVN's system to land.

Learned trust deals with how the certification official interaction has changed over time as they deal with the new technology. In the beginning they may have one opinion, but after seeing the automated or autonomous function perform over time their level of learned trust will adjust depending on the results.

While some level of trust is required for certification officials to accept risk, there is not a clearly defined method for how we build trust in military systems. The current V&V techniques are designed for systems with a qualified pilot or operator. To achieve a level of trust, V&V techniques used for certifying autonomy must be sufficient that the certification officials will have trust in the actions of the decision engines controlling the vehicle, and trust that the system will not pose an unnecessary risk to mission completion.

For a system to operate autonomously, it will need to sense the environment it is currently operating in. Upon sensing the environment, it will then need to properly classify the input its onboard sensors give it. Once it has interpreted the environment, and built its own SA, it will need to perform appropriate actions based on the unpredictable environments it will find itself in. This is a similar process that a fully qualified pilot is expected to perform once leadership has bestowed trust on their actions. One issue with using sensors to build the SA for a UAV is the vast amount of data the needs to be filtered through. Using automation to process onboard imagery is not a new concept. From December 2013 to January 2015 a CubeSat was flown with multiple onboard sensors. An autonomous onboard processing was used to determine what images would be passed back to the ground station for further processing [64].

While trust is required for certification officials to accept risk, there is not a clearly defined method for how we build that trust in autonomous military systems. The V&V techniques used for certifying autonomy must be sufficient that the certification officials will have trust in the actions of the decision engines controlling military aircraft.

2.5 Bringing Autonomy to Military Aviation

It is clear that future military platforms will rely on ever increasing levels of automation and eventually autonomy. To facilitate this, military leadership has taken steps to define investment strategies for implementing autonomy. In a 2011 memo, the Secretary of Defense designated autonomy as one of seven priority investment areas. Shortly thereafter, the Assistant Secretary of Defense, Research and Engineering, (ASD(R&E)), established four working groups to help define the communities of interest. The Autonomy Community of Interest TEVV Working Group was made up a collation of the willing. Membership included all of the services major research facilities and test and evaluation organizations. In 2015 this working group published vital definitions, challenges, gaps and goals (or vital investment requirements) for validation and verification of autonomous systems [8].

- **Challenges**: State-Space Explosion; Unpredictable Environments; Emergent Behavior; Human-Machine Communication

- **Gaps**: Lack of Verifiable Autonomous System Requirements; Lack of Modeling, Design, and Interface Standards; Lack of Autonomy T&E Capabilities; Lack of Human Operator Reliance to Compensate for Brittleness; Lack of Run Time V&V during Deployed Autonomy Operations; and Lack of Evidence Reuse for V&V

- **Goals**: Methods and Tools assisting in Requirements Development and Analysis; Evidence-Based Design and Implementation; Cumulative Evidence through RDT&E, DT & OT; Run Time Behavior Prediction and Recovery; and Assurance Arguments for Autonomous Systems

While these areas have been identified as needing resources and extensive research, as of 2020 they have not been completely solved/mitigated.

2.5.1 Challenges

As the V&V community begin to grapple with how to certify and test autonomous functionality and systems, the state space issue continues to cause issues. Current systems are fielded with a number of test conditions met during V&V process. It is believed that once you put a human operator, or pilot, in the loop they could take the input offered and make a proper decision. This limited the state space required for V&V during T&E. There is an infinite trade space for the V&V community to analyze if we want to ensure autonomous functionality.

In 2015, when examining the technology investment strategy, the Office of the Under Secretary of Defense for Research and Development published the following: "The notion that autonomous systems can be fully tested is becoming increasingly infeasible as high levels of self-governing systems become a reality... the standard practice of testing all possible states and all ranges of inputs to the system becomes an unachievable goal. Existing TEVV methods are, by themselves insufficient for TEVV of autonomous systems, therefore fundamental change is needed in how we validate and verify these systems [8]." While the Under Secretary's TEVV Working Group helped to frame the problem, a solution has yet to be identified.

For tactical UAVs performing missions with limited to no contact with human controllers, it is extremely difficult to determine what actions the UAV will take under various conditions. The mission parameters and inputs will vary depending on the stage of the mission and environment. Moses, Chipalkatty and Platt proposed a belief space hierarchical planning tool to help solve the optimization problem of the immense state space conditions and actions the UAV may take [65]. Their research dealt with UAVs completing a mission relevant task. They demonstrated that completing such a task was more difficult than just going from point A to point B, the UAV had to make mission relevant decisions. They hypnotized reducing the vast state space that this problem presented to smaller subsets of the overall state space. While their approaches do reduce the state space required for ultimate V&V, it is insufficient alone for this research.

Traditional V&V approaches are not appropriate for large software intensive systems where emergent behavior may be considered. The traditional V&V approach, (model the pieces, model the whole, assemble the pieces into the whole and then test the whole) has difficulty with large complex systems. When these systems become so complex it is difficult to model the interactions between the subsystems. Once you add in the effects of the environment and other outside inputs, traditional V&V approaches cannot replicate the possible permutations [21].

One of the current buzz words in the DoD is Cyber. As UASs continue to increase in use and fill vital nodes in the military command and control infrastructure, ensuring the security of these systems against malicious cyber attacks is essential to national interests [96]. Kwon, Yantiek and Hwang developed an algorithm that can detect stealthy cyber-attacks effecting the controls domain of a UAS. The algorithm was designed to work in real time and to make safety critical adjustments [96].

The challenges are not limited to those that have been identified. Ultimately the community needs to identify solutions or mitigation strategies prior to autonomous functionality being certified for DoD use.

2.5.2 Gaps

The gaps identified by the TEVV working group (lack of verifiable autonomous system requirements; lack of modeling, design, and interface standards; lack of autonomy T&E capabilities; lack of human operator reliance to compensate for brittleness; lack of run time V&V during deployed autonomy operations; and lack of evidence re-use for V&V), make it seem as if there is little hope in certifying autonomy. Yet, research has continued to close these gaps.

Health monitoring may be a way to address some of the gaps that have been identified. The inner loop of the control system is where non-deterministic conditions can exist for certifiable systems. The boundary conditions are monitored by the outer loop, or run time monitoring, once the inner loop reaches a predefined boundary condition its behavior becomes deterministic. "System health management is an important feature of autonomy, enhancing consistency checks, overall system robustness and even some degree of self-awareness. Seemingly unrelated, debugging and analysis of such complex systems is another challenge during development that should not be underrated." Torens, Adolf, Faymonville and Schirmer proposed that "the so-called run time monitoring or relevant properties are system requirements is a viable technique to support both aforementioned concepts [35]."

While trying to identity possible methods for certifying autonomous behavior, health monitoring may have a place. One method that has been identified is to define a bubble where the system can operate (outer loop). As long as the systems performance stays within the bubble it can exhibit non-deterministic behavior (inner loop) [37]. When its behavior reaches the edges of the bubble it would exhibit a known behavior. We feel, and naval certification officials agree, that this method offers the extreme promise for certifying autonomy in the near future.

2.5.3 Goals

For a UAS system to operate autonomously, or perform as a pilot would, it needs to be able to complete some basic functions. In the end, these systems will need to accurately sense its environment, build its own SA, and make aeronautical decisions based on current SA. Each time these subtasks are accomplished, the V&V community will be closer to finding a way to certify non-deterministic behavior.

The idea of starting small for certifying various levels of autonomy within UASs is not new. Researchers are the AFRL have investigated certifying aircraft with an adaptive controls via Run Time Assurance (RTA) architecture. The USAF defines airworthiness as the "verified and documented capability of an air system configuration to safely attain, sustain, and terminate flight in accordance with the approved aircraft usage and operating limits [97]." Their approach highlights the fact that certifying autonomous systems is difficult due to the inability to predict the behavior in a given flight condition. By adding a switching mechanism that takes the new autonomous behavior out of the loop once a fault is detected, they feel that certification may be achieved [98].

SA development is one of the key skills of any successful pilot. A pilot builds their SA through their senses. For a UAS to be truly autonomous they must have the ability to build their own SA of the environment they operate in. As vision is one of the most important sources of SA for a pilot, so it is for the TALOS controller of the AACUS H-1 used in this research (see section 2.7.8). When mapping the landing area it uses LiDAR to build a 3D image that it can process to determine a safe landing spot. Using LiDAR for 3D mapping is not unique to AACUS. Researchers as Cal Poly Pomona outfitted a small fixed wing UAV with LiDAR and attempted to build a 3D image from its output. They were successful. Figure 2.8 shows the test area, and Figure 2.9 shows the post processed 3D image of the test area [66].

Figure 2.8: Parado Airfield in Chino Hills, CA, Testing Area ("Google Earth" Image) Used in Reference [66]

There has been numerous academic research done on vision based navigation for UAVs. Agrawal, Ratnoo and Ghose proposed a method for guiding UAVs in an unfamiliar urban environment using image segmentation. "Using the segmented image, the proposed method first identifies the passage between the obstacles, the decision making chooses the closest free passage and obstacle avoidance, and passage-

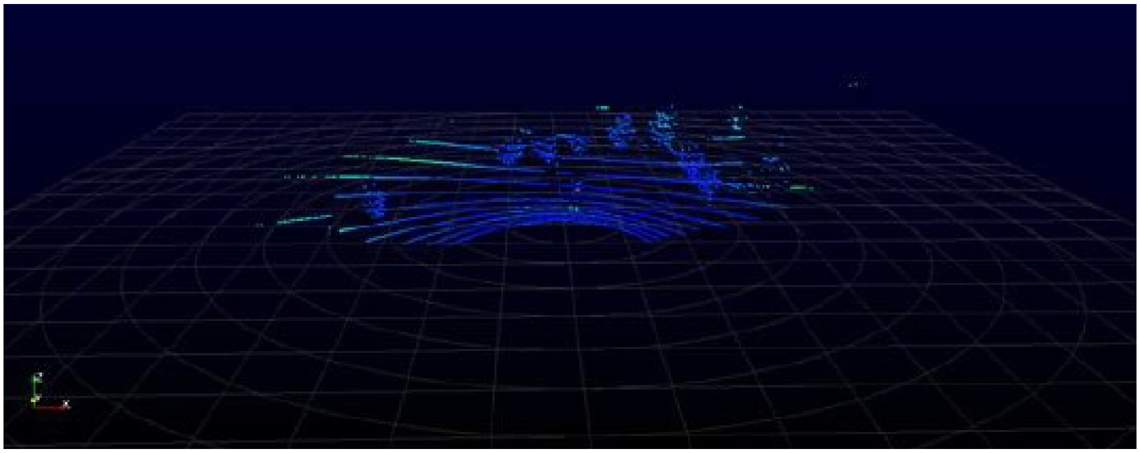

Figure 2.9: Post Processed 3D Map of the Test Area Used in Reference [66]

following guidance law steer the UAV through the passage. Analytical results show a faster obstacle avoidance for the proposed segmentation strategy as compared to existing optical flow-based methods [67]."

Optical flow has often been considered for use in UAV navigation. There have been several academic papers written on the advantages of using it. Through a grant from NASA Chao et al. focused on experimental validation of navigation information obtained in wide-field optical flow, using UAV flight test data. They determined that "optical flow information contains accurate enough navigation information that could be used for UAV applications [68]."

Linear controllers are relatively easy for UASs. It is easy to predict the various functions that a control needs to accomplish under predictable linear conditions. The issue comes when you have a requirement for a non-linear controller. Traditionally aircraft have had a well-trained non-linear controller (a qualified pilot) as the state of the art moves toward autonomous flight vehicles, we need to find a way for the vehicle to be its own non-linear controller. Novak and Bhandari, from Cal Poly

Pomona, demonstrated the ability to train a neural network to be a non-linear controller of a small RC UAV [75].

2.5.4 Software Development and its Implications on Certification of Autonomy within Naval Aviation

The systems design concept is prevalent in all DoD acquisition systems. To simplify the various steps of the process, the classic V model is often used. The left side of the V can be considered the coding or development phase. While the right side can be considered the certification side. The TEVV working group used the V diagram to describe the various steps in the V&V problem as well as their interrelationships (Figure 2.10 & 2.11 [8]).

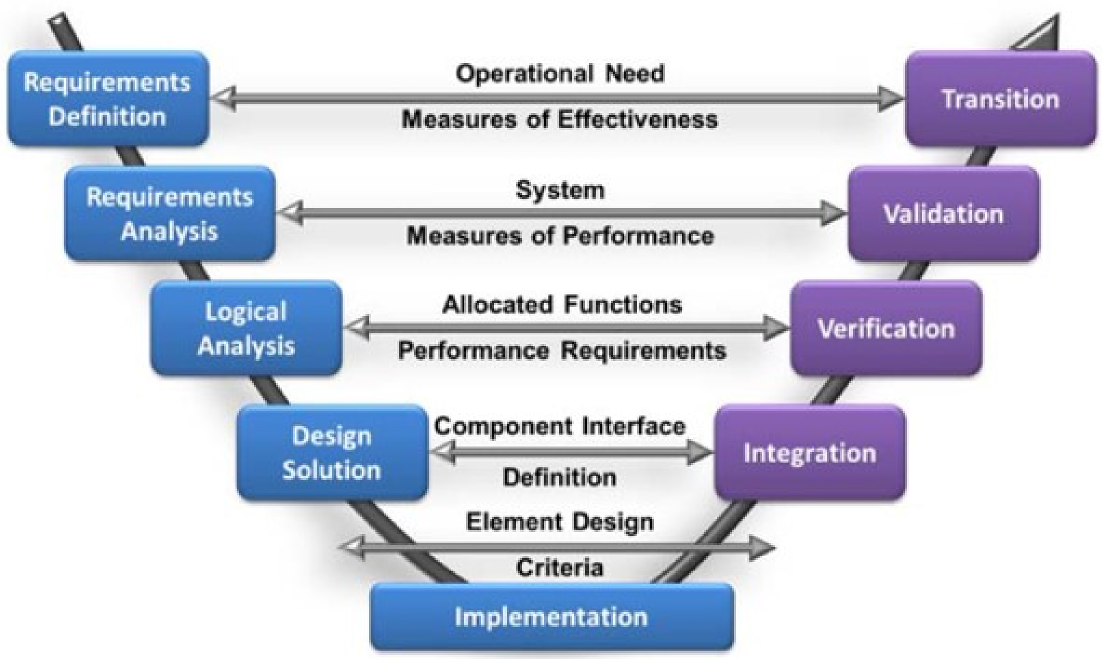

Figure 2.10: Classic V Development Cycle [8]

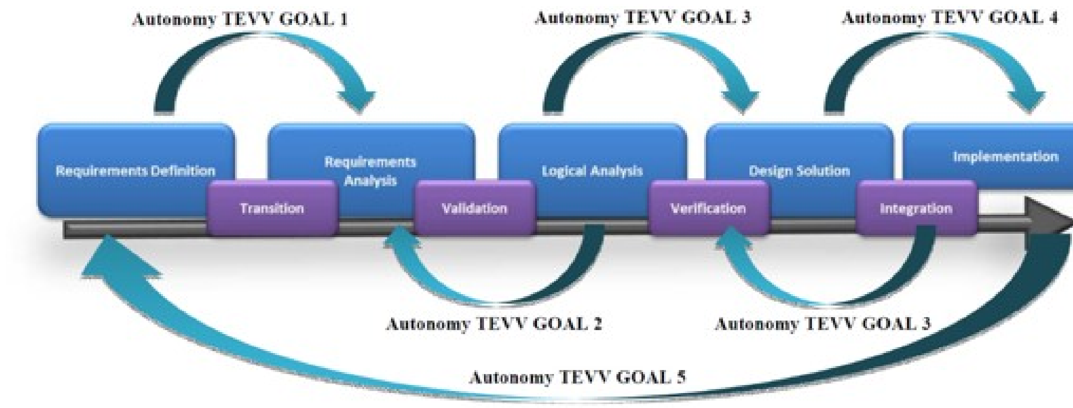

Figure 2.11: Autonomy TEVV Process Model, Integrated with the Traditional [8]

In a comprehensive review of current V&V strategy of autonomous systems Fisher, Dennis and Webster detailed how systems are currently analyzed based on the varying levels of autonomy and the systems direct interaction with the environment. They summarized three categories of autonomy:

- Agent (high level reasoning)
- Control (the decision engine that would make decisions)
- Hardware (the part of the system that interacts with the real world)

Their summary can be found in Figure 2.12 [24]. Their summaries are valid based on current use of autonomy (in areas where direct control of robots is not possible, or dangerous). However, it is lacking the element of accountability required for military systems. An example would be the current application of autonomous systems in toxic environments. If a system made an incorrect decision, the greatest risk would most likely be limited to the loss of the system. In military UAS applications, a qualified pilot, or operator, is trusted to control a vehicle where an error may lead to an international incident.

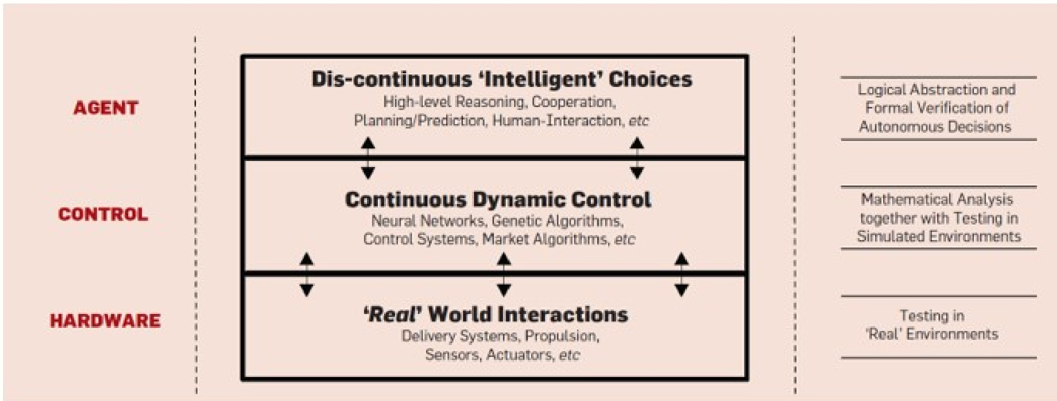

Figure 2.12: Typical Hybrid Autonomous System Architecture – with Suitable Analysis Techniques Noted in Reference [24]

In recent years the development cycle for new avionics systems has been compressed. This compression has necessitated new ways in which we perform V&V of new software. Abraham was able to summarize the use of the V model for the various steps in V&V. A V model is often used in systems engineering. Figure 2.13 is a software development V Model. It begins at the top left with system requirements, then high level design followed by detailed design. Once a detailed design is decided upon, the coding stage can begin. The next step is unit testing and integration testing. Providing a software can pass all of these steps it is considered verified and validated. However, the earlier an issue can be identified the less resources (both time and expense) will be required to correct the defect. To do this Abraham recommends using software based models for V&V during all phases of development to include the left side of the V model [20]. This technique will be vital for V&V of military systems. The resources devoted to V&V and T&E are limited, and have a tendency of being reduced during execution. The earlier a defect can be found the better.

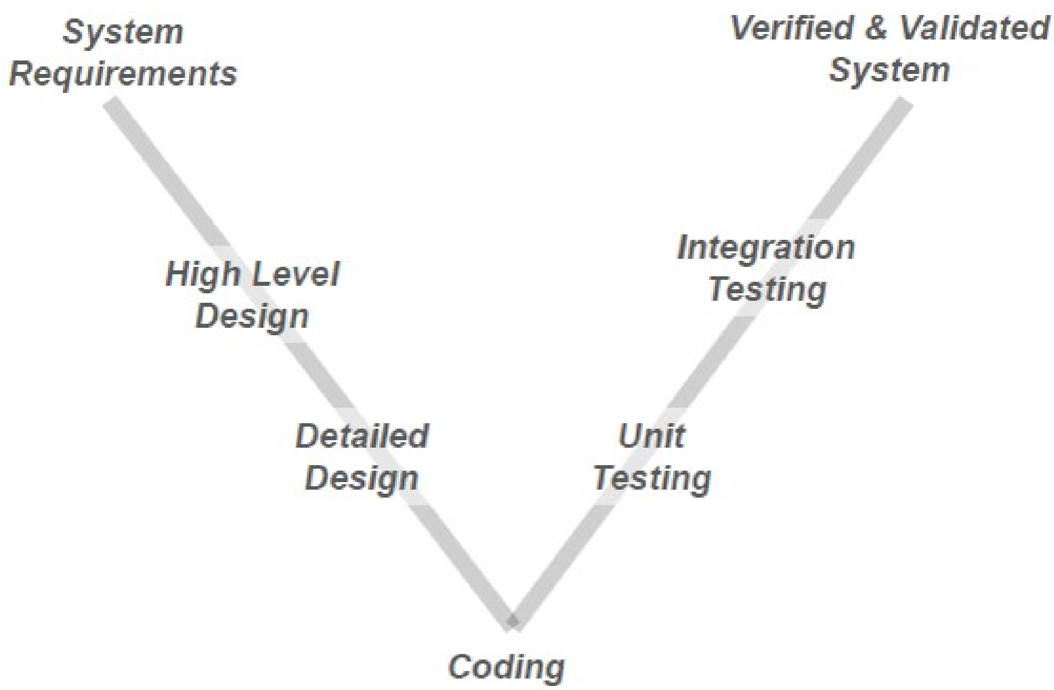

Figure 2.13: V-Model as Described in Reference [20]

Modern civilian aircraft (such as the Boeing 787) are extremely software dependent. In an attempt to standardize V&V in these aircraft the FAA approved AC 20-115C on 19 Jul 2013, making DO-178C a recognized "acceptable means, but not the only means, for showing compliance with the applicable airworthiness regulations for the software aspects of airborne systems and equipment certification [3]." DO-178C outlines approaches to have tractability in the V&V process that maps requirements to systems performance. This tractability requirement is similar to the artifacts that naval flight clearance authorities will require prior to certifying autonomous naval aviation systems.

The last 60 years has seen an explosion in the amount of software installed in military aircraft [21] (see Figure 2.14). As the military moves toward automation taking over tasks currently reserved for pilots, the amount of software functionality will increase.

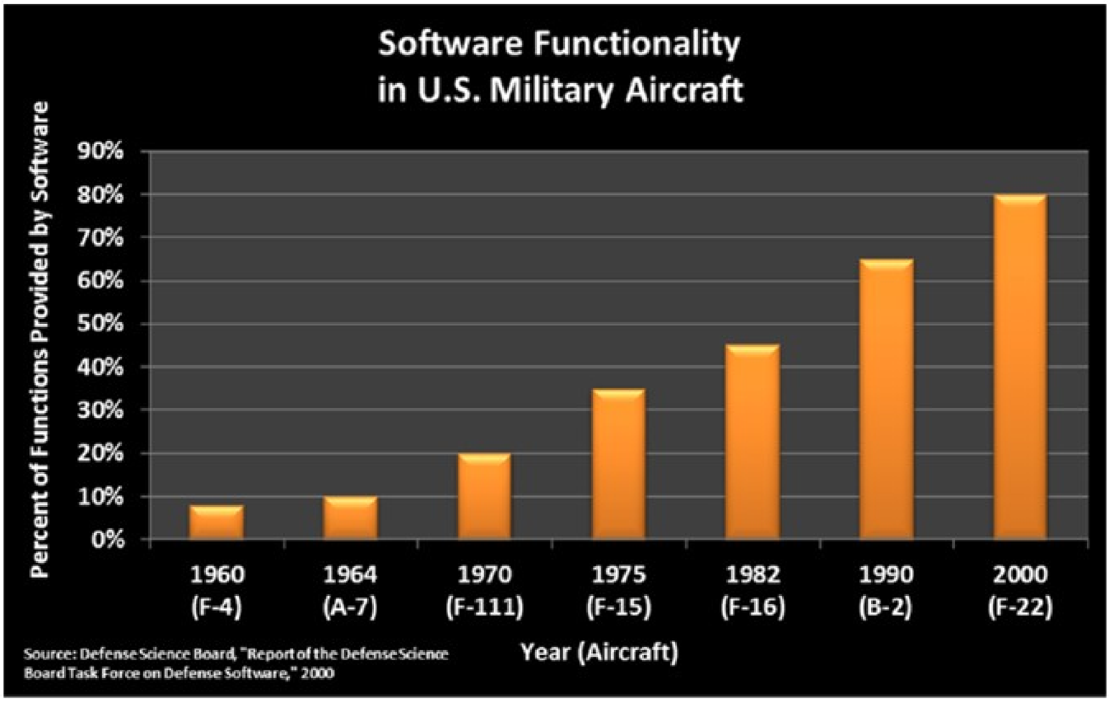

Figure 2.14: Increasing Aircraft Functionality Provided by Software [21]

In 2016 Eiemann and Allan demonstrated how to use various software tools (e.g. Simulink/Targetlink) to be in compliance with DO-178C whose development phases, including the required verification steps, are shown in a simplified form in Figure 2.15 [26].

Emergent behaviors are a difficult problem for the V&V community. As these behaviors are unpredictable by definition. There is limited ability to reproduce the behavior and what may have led to it [99]. "Thus, a potential means of dealing with such a V&V challenge is oriented more towards tolerance during operation than detection during development. The concept of resilience, as applied to systems and

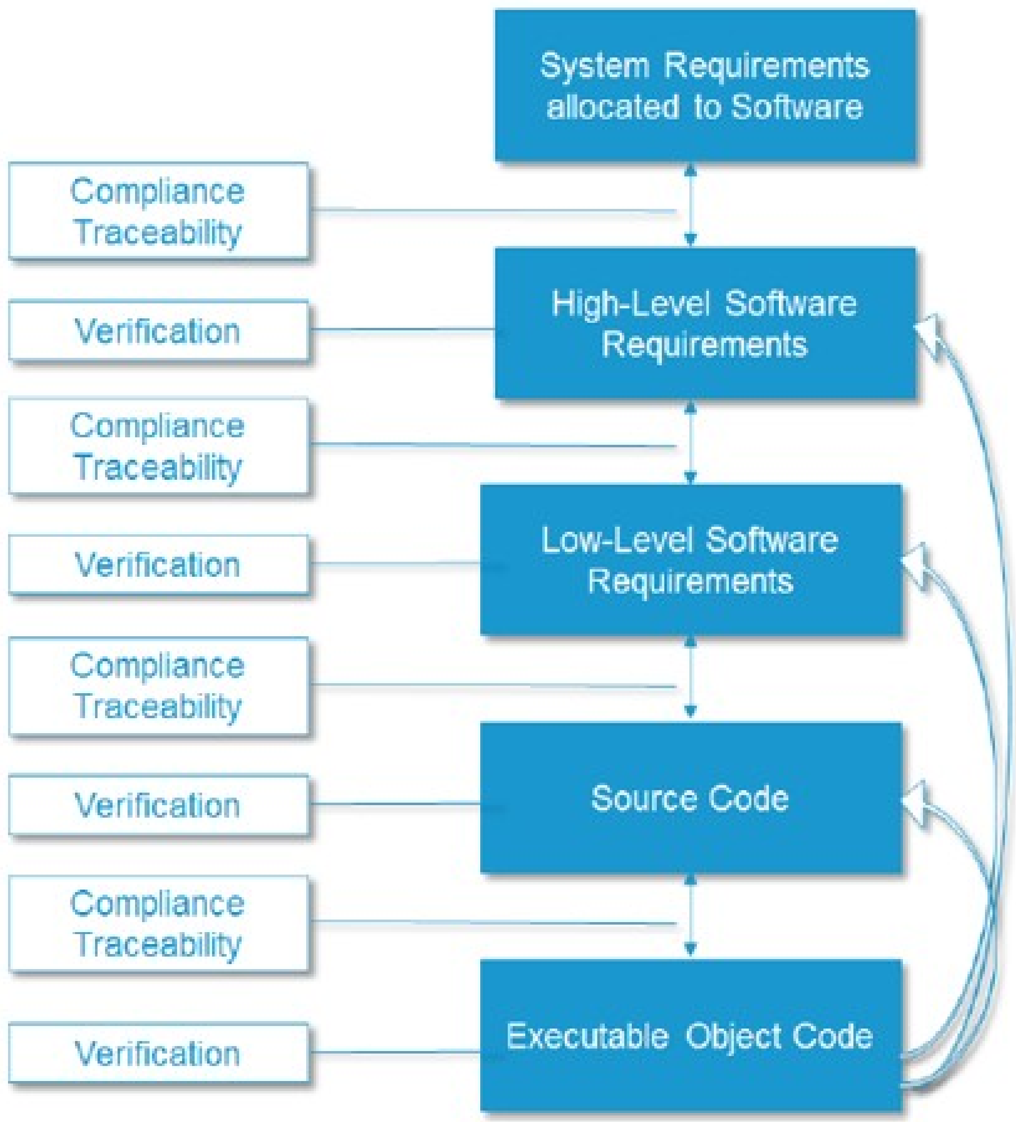

Figure 2.15: Important Development Phases According to DO-178C, Including the Necessary Verification Steps (Architecture Design and Verification have Been Omitted) [26]

software, is exactly that: Designing systems and software to detect the unexpected during run time, Adapt or adjust as necessary, and continue functioning (albeit potentially in a degraded mode) [21]."

For policy authorities to actually authorize software, or a decision engine, to control critical portions of an UAV there will be extreme scrutiny on how the software was developed and how the requirements definition was architected. In 2014 Walker, Shan and Liu pointed out that “software framework is constantly being designed form scratch over and over again, and there is little to no discussion regarding how it was actually designed [28].” This is not helpful when numerous UASs are currently being developed for integration into the national airspace system without a standard for which they are to be certified to. Without a standard for software development, certification officials will have a difficult risk decision. There needs to be a standard for how flight critical software is designed and coded.

When dealing with complex software, it is extremely difficult to determine the best way to test to ensure it is functioning properly. This is especially true when dealing with unproven software that will be responsible for taking actions that are currently reserved for aircrew in the autonomous system. NASA has long been a front runner in defining leading edge technologies for aviation. When designing their Trick Simulation Toolkit the identified “Understanding of the requirements of testable software, test automation tools, and adoption of the Test Driven Development Process [100]” as items that dramatically improved the testing toolkit.

When it comes to certifying new software for aircraft, the use of M&S is key for reducing the cost and scope of flight test. In order for this reduction to take place it is vital for the system requesting certification to show that when hardware is in the loop, it performs the same as it did when it was simply a simulation. Otherwise, the simulation model will be invalid. An example of this was complete by researchers at San Jose State University. In 2015 they showed that you can use MATLAB based software for low cost COTS programming of a UAS dedicated for autonomous flight testing and control system design [101].

2.6 Formal Methods Relation to Naval Aviation Automation

In the words of the former chief engineer of the USAF: It is possible to develop systems having high levels of autonomy, but it is the lack of suitable V&V methods that prevents all but relatively low levels of autonomy from being certified for use [93]. The AFRL funded a study asking a question regarding the state of possible processes for certification of UASs which employ machine learning or autonomous functionality through some sort of evidence based licensure process. These categories were: Formal Methods; Requirements and Metrics; Normative Oracle Generation; CoActive Design; Implications of Learning Autonomous systems; and M&S considerations for licensure of autonomous systems [94]. In the near future certification officials will be asked to certify autonomous systems. Until the V&V community can develop solutions to these issues, officials will be reluctant to accept the risk these new advances offer.

Some certification authorities are requiring all possible states of an autonomous system to be tested to verify how the system will function. The sheer volume of these conditions make this resource prohibitive (both in time and financial cost). It is also a well-documented fact that the earlier in a systems development a defect can be identified the fewer amount of resources will be required to fix the defect. Gross et al. identified these factors and proposed using formal methods applied to identify issues early in a systems design [19].

While formal methods is a broad topic, three of the most promising techniques for its use in certifying autonomous systems are with autonomous systems are: Model Checking; Theorem Proving and Run Time Verification. As part of his doctoral research from Carnegie Mellon University, Berezin summarized model checking and theorem proving in relation to the verification of software intensive systems [44]. Kane's doctorial research (also form Carnegie Mellon University) summarized run time monitoring for safety critical embedded systems [32].

2.6.1 Model Checking

Model checking is an automatic technique that can only be performed on finite state systems, many are expressed via finite state transition diagrams. It involves developing simplified models (in mathematical terms) which captures the essential features of the system (not the entire system). The specifications (derived from the requirements of the system) which it is to be verified against is normally expressed in terms of logical statements. Following the simplification of the system, and the definition of the specification, a software tool is used to perform an exhaustive exploration of the state space. This is one of the limitations of this formal method with respect to certification of an UAV making decisions normally reserved for a pilot or operator. Depending on the degree of the simplification, officials will most likely be reluctant to rely on this verification technique based on the amount of risk this will necessitate during certification. The advantages of model checking are: In contrast to theorem proving, model checking is completely automatic and fast; It can be used to check a partial specification and can provide useful information about the correctness of the system even if the system has not been completely captured within the model.

An extensive overview of model checking can be found in Baler and Katoen's book *Principals of Model Checking*. [31]. In their book, they point out that model checking is an automated technique that can can be used to ensure system is free from errors. As systems have become more complicated, model checking methods have advanced to a state where they can be considered mature and used as a valid technique for verification and debugging purposes [31].

Bakera et al. presented a game based model checking technique for safety critical actions of the ExoMars Rover. Their work focused on the actions that the rover would take, and the various branches the actions would led to. Model checking is used to decide whether an abstraction of a reactive system satisfies a requirement. They used model checking via a parity game based approach to show a remote/autonomous system (such as a Mars Rover) would not get into situations where it lacked the programming to recover. They showed that during the game based model checking, both winning and losing situations would reveal meaningful information to designers. Generally speaking, the paper covered an interesting use of model checking. The model the authors proposed included the rover, the martian environment, and the actions the rover would take. It is unclear how precise the model itself was. We can see that using model checking to ensure safety critical actions is a viable method. However, if the model itself is not verified (tough to do in this case as there probably is not enough information to completely characterize the environment of the rover) the use of a model checking would not effectively provide enough data for the clearance officials to accept the risk of its use. The use of a “Game-Based” approach is an interesting idea, and would make it easier to analyze the results of the model checker [47].

Webster et al. presented a “proof-of-concept approach to the generation of certification evidence for autonomous unmanned aircraft based on a combination of formal verification and flight simulation.” Their work was is an attempt to help in the certification of UAVs to operate in the national airspace system. As with all model checking, the *model* is a mathematical representation of the system and its interaction with the environment. The purpose of the model checker is to ensure that under all situations the model will not violate a safety critical requirement. Their work sued the Java PathFinder (JPF) tool developed at NASA Ames Research Center. JPF allows for both deterministic and non-deterministic behaviors (required for the uncertainty of autonomous functionality). As with all models, the better the information used to create the model the better the model. The more detailed the model, the better the results (and the higher the cost in resources). In general their approach is simple... code the actions a pilot would take and the environment interactions into a model. Check the model for possible safety critical interactions. Once the model checking is complete, perform M&S to reduce the risk of flight test which will eventually lead to a flight clearance. While this appears to be a good approach, the fidelity of the information that is used to build the model is the key to its correctness, and eventually the utility of the output of model checking [5].

Sirigineedi et al. points out that if the system contains discrete-events it can be modeled by a finite state graph it will be suitable for formal verification by model checking. Essentially, break the system down to a simplified set of logic statements. The model checking tool can then go through all of the situations to verity it will meet the specification (developed form requirements of the system). Their paper is an overview of model checking procedures. Again, the robustness of the model is the key to its utility. The more simplified it is (easier it is to develop) the less utility it will have for certification officials [48].

Webser et al. presented a paper in 2012, that was a precursor to their 2014 Journal article. They presented a method for model checking quantitative requirements (such as the actions pilots would take). They pointed out that any evidence gained from model checking would need to come from a verified tool to be of any use to certification officials [45].

Humphrey explored using model checking for the verification of the VIP escort mission. The mission was simplified, and all points could be defined. This is an ideal use of model checking, there is limited "branching" to behaviors that cannot be anticipated. One limitation was the lack of dynamic events (such as a reroute during the middle of the escort) within his simplified model [49].

Verzino, et al. had a different use for model checkers. As with other approaches, they broke down the requirements to mathematical basis and then ran the various permutations of the system. Yet, unlike other approaches they did not reject a failed state. It was those failed states that drove further simulation. This is an excellent approach, as when the model checker found an issue it did not revert to a failed state (the model itself may be failed). M&S can then be used to analyze these potential failed states to see a more detailed response [50].

Humphrey and Patzek break with past uses of model checkers. In the past the checker would be used to find faults with the system and how it is designed to comply with a specification. In their work they proposed using the checker to see if the system could complete a requirement or task. They exercised their proposal within the ISR domain. The paradigm resembles the Observe, Orient, Decide, and Act (OODA) loop common in military pilot training [51].

Torens and Adolf point out that safety is a primary concern in the aerospace industry. This includes the software that is used in aviation systems. The metrics associated with software development can be found in DO-178C. Their work actually breaks down how formal methods, and model checking in particular, complies with DO-178C [52].

Hansen et al. had a solution for one of the drawbacks of model checking (when the system is to complex for the traditional formal methods approach). Statistical model checking is a useful tool for evaluating software systems operating in stochastic environments. They also used sampling to reduce the simulation requirements in areas of limited failure rates [53].

Knowing the mode the aircraft is operating is vital to the action taken by the aircrew. If there is a confusion, incorrect fight control inputs may be input and lead to a mishap. Flight deck mode confusion detection has been a recurring problem studied by the research community for the last few years. Nandiganahalli, Lee and Hwang's 2017 paper broke down an actual incident into a stochastic linear hybrid system model that can input into a model checker to ensure it meet the requirements [54].

2.6.2 Theorem Provers

Ghorbal et al. advocated using theorem proving in addition to classical V&V techniques for software intensive systems. As software takes over critical tasks in modern aircraft, there needs to be assurances that is will not violate safety critical boundaries. While theorem provers offer the ability to verity a system will not violate a safety critical boundary, this paper details several challenges of using theorem provers for aviation systems [102]:

- Uncertainty (difficult to predict the all situations the system will eventually operate when you take uncertainty into account)
- Proof automation (unable to fully automate the process)
- Numerical issues (computers cannot effectively perform real number computations
- They are truncated to fit the finite representation
- Scalability (looks at small pieces, rarely at the whole system)

Coutieu et al. used Coq (a theorem prover similar named for its developer Thireey Coquand) to show a behavior is possible for an undefined number of autonomous robots. To simplify the decision space they made a number of assumptions. Their work is adequate for theoretical research, but it has limited utility for real world systems. Some of the challenges addressed by Ghorbal et al. are highlighted in this work [103].

Jiang et al. point out that model checking and theorem proving are similar as the both are based on decomposing the system and then applying a number of rules to system for the verification process. They offer several contribution that blend model checkers and theorem provers. This work is still fairly theoretical, and needs to be matured before it can be adequately used for certification. It does not address how it will deal with the disadvantages of both methods as they combine [59].

Asokan et al. presented a method for automating how they used the PVS theorem prover. One of the issues when dealing with theorem provers is the fact that the programming language it uses for theorem is specific to the prover. This research pointed out the need for theorem provers for safety critical software functions [60].

A technical report from MIT provided an excellent overview of model checkers and theorem provers. In the conclusion of the paper it points out: Because theorem provers and model checkers each provide complementary benefits in terms of automation and scalability, it is likely that this trend will follow and the model checks will continue to be useful on systems of manageable size while theorem proves will be used on large systems [55].

Sutcliffe et al. echoed many of the points that have been discussed in other papers. Since aviation software is completing more and more safety critical tasks, certification officials need to be assured that the software will comply with safety considerations. Their paper described and evaluated a semantic derivation certification approach to proof checking, the evaluation of which is the papers main contribution. They highlighted a number of safety obligations that the software would need to comply with for aeronautical certification. Their method was able to verity 129 out of the 131 proofs identified, and provided traceability required by certification officials [56].

Goodloe, Gunter, and Stehr work is not aviation related, it is theorem proving related. They came up with similar conclusions to those that have been known in the aviation software development community. The sooner you find an issue, the easier it is to fix (and the less resources it takes to fix the issue). Their work was focused in wireless network protocols. There work deals with using theorem proofers to develop the protocols. They showed that by doing this, there will be less of a need to fix the protocols that are developed as they will have less defects [58].

The NASA Langley formal methods group is one of the leading government research centers focused on certification autonomy in aeronautics and astronautics. They have been at the forefront of software certification to enable autonomous systems to complete tasks currently reserved for pilots. For certification officials to approve this they will need a way to certify that the systems will not violate safety critical boundaries. One of the areas NASA is focused on is path planning. In a paper focused on their Airborne Coordinated Conflict Resolution and Detection (ACCoRD) framework program, they detailed the conflict detection algorithm for two aircraft flying polynomial trajectories. The algorithm used was derived from theorems that were proved in PVS. This shows a method to build algorithms based on verified theorems [104].

The NASA Langley formal methods group is working on (and have published) several articles on obtaining flight clearances for UASs to operate within the national airspace. Their use of “Theorem Provers”, or “Proof Assistants” has led to the definition of a “well clear” volume for UAS operation. This is being used to help certify UASs for the “see and avoid” requirement [72–74,105]. This work, and several others, highlight the use of proof assistants to verity the requirements placed on the software. The algorithms that are developed for the system contain steps from the verified theorems [57].

Muñoz presented the work the NASA Langley formal methods group has been researching for UAVs certification for operations withing the national airspace system. In the lecture he covered the formal methods approach used by NASA for verifying the algorithms used for eventual certification. He also pointed out the challenges that PVS (theorem prover) has when dealing with cyber physical systems. These challenges will be similar to the ones we faced in our research [61].

In an attempt to provide evidence that a UAS can operate in the national airspace, Narkawicz and Muñoz used a formally verified conflict detection algorithm to establish how a vehicle would react when operating on a non-linear trajectory. This work was just another example of the NASA Langley Formal Methods Group attempt to move the certification of autonomy forward for the *well clear requirement* [105].

Narkawicz, Muñoz and Dutle published a paper where the use of onboard systems (such as TCAS) to help determine the actions of a UAS when operating in congested airspace. They presented a formally verified approach for coordination of aircraft maneuvers to avoid collision [74].

2.6.3 Run Time Assurance

One possible method for certifying an autonomous system to perform tasks currently reserved for qualified pilots was detailed by researchers supporting the USAF. Gross and her team examined the use of run time assurance and formal methods analysis for non-linear system control.

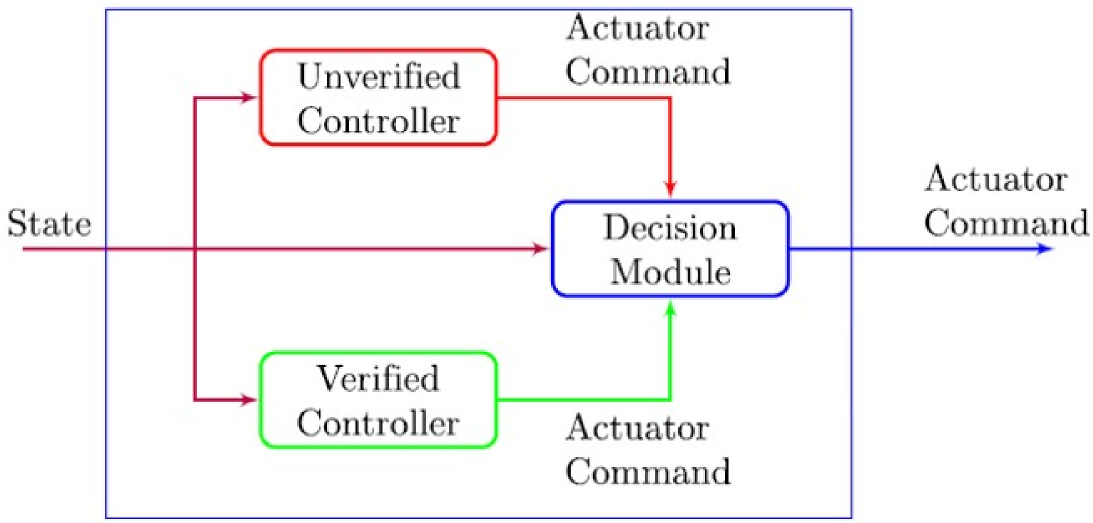

Figure 2.16: Run Time Assurance Architecture as Described in Reference [36]

Schierman et al. proposed, and demonstrated through M&S, that a run time monitor can be used for UAVs to protect the vehicle from unsafe situations. They pointed out that as systems become ever more reliant on software it is reaching the limit of current V&V techniques. They refer to it as a safety wrapper. While operating within the safety wrapper the primary controller controls the actions of the vehicle. Once it reaches the wrapper, the fail-safe or backup controller takes over. Their approach is basically a band aid for the lack of V&V capabilities possessed by certification officials [37].

Lichter et al. presented the use of a run time monitor in flight test. As an overview of their approach, the Run-time Observation-based Margin Estimation (ROME) software toll was tested onboard the NASA Langley's AirStar Test bed. The ROME tool is designed to reduce the risk for flight test of advanced control laws. By having a software tool that can counter unsafe actions in flight test is an impressive risk mitigation tool [38].

Rabideau et al. approach was geared toward space operations (as it was presented at the SpaceOps conference). Space platforms have limited computation ability, and may have limited interaction with earth based resources. Their paper dealt with using a run time monitor to help prioritize the limited autonomous computing power. Their algorithm is designed against typical spacecraft operations scenarios [39].

A paper from Oakland University (Rochester, Michigan) demonstrated run time monitor research can be used to develop, manipulate and test changes at the task and parameter level. In addition to the changes the monitor could make to the system, it provided feedback to the designers on numerous parameters via a graphical interface. While not a traditional run time monitor in the formal methods arena, their approach enabled them to monitor the system during changes to the software during test. Their approach was ideal for any developmental flight test [40].

Aiello et al. point out a fact that is well known. Control research has made dramatic increases in autonomy, but V&V techniques have failed to keep pace. Traditional techniques are based on proving the entire state space for what the system will do. Their research dove tails with ours, as we propose to demonstrate what an autonomous system will not do. They recommended a run time monitor to prevent systems from violating a clearance envelope. Their approach is similar to the safety wrapper they previously published. Their work details that a run time monitor can provide risk mitigation for new control laws during flight test [41].

Wong et al. presented the use of run time monitors for advanced propulsion systems. The theme of current V&V techniques used for certification can be considered inadequate for new software and hardware combinations. It allows the new system to operate until it hits a limit (or anomalous behavior) considered safety critical. At that time it would revert to a simpler certified system [42].

Researchers at AFIT (along with some outside researchers) detailed how using a run time monitor can be used for satellite control. They detail that emergent behaviors cannot be verified at this time. But if used within a safety container (similar to the wrapper previously discussed) they can be used, providing there is a certified control method at the limits of the container. They then used formal methods (model checking) to show the run time monitor would provide the safety limits. The main limitation of their approach is the simplification assumptions taken in the analysis [34].

Huang et al. presented a different run time monitor. It used machine learning to monitor the output of aviation software. As most approaches to formally verify software are at the sub component level, the interactions of the complex systems are not always properly analyzed. The approach in this research developed an algorithm that was tested in M&S to see if it could meet the standards of aviation reliability. The paper details a limit of formal methods currently used in the field, the interactions of the complex systems creates bugs that are not always accounted for [106].

Avram et al. research was in support of the UAV mission within the USAF. They proposed using a run time monitor that can account for failures within a quadrotor during nonlinear adaptive control testing. Their algorithms would shift the control from an uncertified nonlinear to a certified linear controller when faults were detected. This is an example of the use of a run time controller switching from uncertified to a certified system when safety limits are reached [43].

Dillsaver et al. research was focused on using an uncertified controller within a clearance envelope. If the controller reaches a threshold of safety or unanticipated behavior it would shift to a certified controller. Their focus was an attempt to use adaptive controllers and stay compliant with military standards for certification. They also performed flight test using quadrotors [107].

2.7 Other Topics for Autonomy in Military Aviation

2.7.1 Requirements and metrics

"Intelligent control designs based on artificial intelligence and machine learning promise superior performance over traditional control techniques; however, the lack of transparency in intelligent control systems and the opportunity for emergent behaviors limits where these system may be applied. Run Time Assurance (RTA) is a proposed methodology to allow intelligent (unverified) controllers to perform within a predetermined envelope of acceptable behavior. Rather than depending entirely on offline verification, RTA provides an online verification approach. Based on the simplex architecture, RTA architectures use a decision module to monitor control systems performance and switch control form an unverified controller to a verified controller if the unverified control violates acceptable behavior ranges or in forced to operate outside of predetermined conditions [34]."

Sankararaman and Krishnakumar described decision making frameworks for UAVs under uncertainty. The idea behind their research was to allow the system to remain in a safe condition as it encounters uncertain conditions (see Figure 2.17). Their paper presented "a computational framework for decision making under uncertainty, to facilitate the autonomous, safe operation of small drone-like unmanned aerial vehicles. This predictive framework was based on the identification risk-factors that affect the safe operation of such vehicles, and predicts the occurrence of events related to such risk-factors during the operation of the vehicle. By analyzing various risk-factors, the framework classified possible trajectories into four categories [108]":

- Nominal and Safe
- Off Nominal But Safe
- Unsafe and Abort the Mission
- Unsafe and Ditch the Vehicle

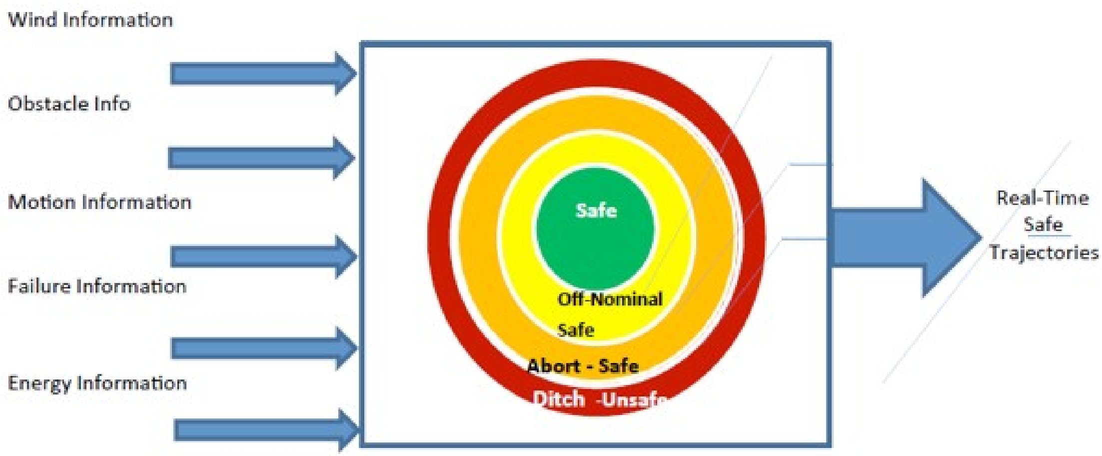

Figure 2.17: Goal of Decision-Making: Identify Safe Trajectory [108]

2.7.2 Normative Oracle Generation

Cowlagi and Sperry examined simplified UAV guidance control based on a cost function. They described "classical planning problems", have the UAV pass multiple points, in an effort to illustrate that there may be multiple methods for the UAV to autonomously visit all of the waypoints assigned. They then developed an algorithm that was able to determine which unique path offered the lowest *cost* to complete [71].

While midair collisions are rare for manned aircraft, they still occure. Jenie, Kampen, Ellerbroek and Hoekstra used monte carlo simulations to illustrate the conflict detection and resolution system programmed into UAVs while operating together can enable them to de-conflict from each other. Their simulation was done for the most stressing case (2D, in a heavy traffic area) in order to force the maximum number of conflict resolution conditions [62].

"Planning and information gathering algorithms are typically based on normative models for reasoning under uncertainty: An autonomous agent seeks actions that maximize some expected utility, given models of uncertainty and specification of cost for some set of tasks/subtasks. This approach can lead to extremely sophistication non-deterministic behaviors through hierarchical reasoning, and provides a flexible means for coping with imperfect information. However autonomous reasoning ultimately depends on several key pieces of knowledge that subject to their own uncertainties which could potentially be mitigated by interaction with human collaborates. Of particular interest are uncertainties in: (i) world models (i.e. imperfect knowledge of possible outcomes that may develop in a particular operating environment): (ii) capability of an autonomous agent; (iii) information sources (e.g. sensor data for own state or task/world state, intelligence reports, ect.)... Yet these approaches to analyzing and certifying autonomy require considerable offline computational effort to exhaustively root out failure modes or exceptional scenarios that are not anticipated or easily understood by system designs are end users. They are also very sensitive to changes in system architecture, mission requirement or uncertainty specifications [109]."

2.7.3 Coactive Design

Ashokkumar and York discussed the use of controllers for unmanned vehicles in combat. While it may be ideal for these vehicles to be controlled by human operators on the ground, combat conditions and the amount of unmanned vehicles in use will most likely make this condition unattainable. This will lead to the requirement for some form of autonomy to be employed by the UAVs. Researchers at the USAFA recommended using a controller that can perform the needed autonomous functions to be based off a nonlinear model of the UAVs. Their controller was designed based on the linearized model of the nonlinear aircraft whose Jacobian matrices are evaluated for a trim (or equilibrium) point [110].

Humphreys, Gobb, Jacques and Reeger are researchers associated with the AFIT. In recent years the idea of having some form of autonomy in tactical fixed wing platforms has taken hold among the leadership of the DoD. The most likely first step in this is the idea of a "Loyal Wingman". The rough concept of employment for a "Loyal Wingman" would be through Manned-Unmanned Teaming (MUM-T). in MUM-T a fighter sized UAV would be paired with a manned fighter. The manned fighter would give missions to his attached UAV to complete autonomously prior to returning to its manned wingman. Through simulation AFIT researchers showed that "the optimal control problem and multiple scenarios are established for a static, deterministic threat environment, additionally, a dynamic and measurement update model are established for raking and successfully avoiding dynamic, non-deterministic threats. A first set of results demonstrates a loyal wingman and dynamic route re-planned algorithm in the midst of op-up stationary threats and a changing mission rendezvous requirements [92]."

Modern aircraft are extremely complicated. The amount of information available to pilots would baffle aircrew of fifty years ago. In some cases the amount of information is overwhelming. The abundance of inputs available to aircrew has lead to increasing level so automation of onboard aircraft systems. "The automation system has been introduced into the cockpit to help the pilots with the operational accuracy and efficacy. However, the automation-centric design has lead to a new safety concern called human-automation interaction issue: Where the expectations of the pilot run contrary to the behavior of the automation systems. The detection of this dysfunctional interaction between the pilot and the automation system becomes important and challenging since it may cause severe aviation accidents [111]."

NASA has done extensive research on the unmanned V&V process. In January 2017 Brat described the current progress they have made for V&V of flight critical systems. "In this paper, we have described parts of the work done by NASA to address the high cost associated with current V&V processes in civil aviation. We have described many tools that can be applied at early phases of the lifecycle, thus enabling to catch errors closer to where there have been introduced. We believe that a systematic application of our tools will enable industry to reduce their cost by avoiding catching errors late in the process (at testing or even acceptance testing), which yields additional re-design or recoding costs [112]."

2.7.4 Implication of Learning Autonomous Systems

In an effort to lead turn the impending need for certification of autonomous aircraft which employ machine learning, the AFRL has been struggling with how to certify these aircraft once they are delivered by industry to the military. In 2016 AFRL received the final report from the Institute for Defense Analysis (IDA) for Project AK-2-3944 "Pedigree-Based Training and Licensure of Autonomous Systems". In the report IDA identified several deficiencies in using traditional test approaches for autonomous systems which employ machine learning. They identified that unlike current approaches, constant monitoring of the performance of the system is required throughout the lifecycle of the system, not just during initial developmental and operational testing. IDA also identified several Science and Technology (S&T) investment opportunities that may overcome the shortcomings [113]. "A key requirement for the current generation of artificial decision-makers is that they should adapt well to changes in unexpected situations [23]." They looked at the possibility to tweak the parameters of an AI so that it can be used as a training tool in simulation for training pilots in "dogfighting".

In a 2015 paper, Junell, Ban Kampen, Visser and Chu found "The fields of automation and machine learning are largely benefiting from the rapid development and availability of computing power everywhere and any time [114]." However, as with most technical reports, they did not focus on the certification question. If you remove a human from the equation, who bears the responsibility of the actions of a system using machine learning.

Emergent behaviors are the future in UAS control. Someday droids similar to those in the Star Wars franchise will pilot aircraft that carry out combat missions or ferry personnel from point A to point B. Yet, there is a need for studying these emergent behaviors and showing that they can be successful. In 2014, Junell, van Kampen, de Visse and Chu studied using "a reinforcement learning task for a quadrotor in an unknown environment. By learning form interactions with the environment, this learning approach works towards more adaptive and robust control laws for autonomous MAVs... This work is just one step toward more autonomous quadrotor flight. Further research will look into challenges of working in the continuous domain, onboard reward or state recognition, adaptive learning for changing environments and use of learning algorithms for improvement of inner loop control for multiple platforms [115]."

2.7.5 Modeling and Simulation Considerations

It is clear that simulation will be key for certifying autonomous systems. The number of actual test points required to validate every possible flight condition is cost prohibitive. Yet, for certification authorities to accept the simulated data in place of actual flight test, the models have to be validated. Tobian and Tishler examined stitching together multiple facets of the flight envelope of a business jet to simulate a continuous model of the flight envelope. They then had qualified pilots fly the simulation and validate it was an accurate example of the aircraft [25]. This method, while limited in its application, may be an appropriate method to gather data for eventual certification via a simulated environment.

The key to a valid aircraft dynamic model is to have the correct aerodynamic coefficients, stability and control derivatives and various constants associated with the governing aerodynamic equations. However, there are many times that this is not possible during aircraft development. After an aircraft is fielded these values can be inferred through various flight test data. Kamal, Bayoumy and Elshabka described a process of obtaining these variables to tune the simulated model of an

RC aircraft [116]. Yet, these fight test data points may be catastrophic. In 1993 an S-3 Viking crashed during one of these test events. The Viking's mission was to use rudder doublets to help excite dynamic modes. However, these doublets ended up exceeding the structural limits of the aircraft and it crashed, both test pilots ejected safely [117].

The words of Box echo today in the M&S world: "All models are wrong but some are useful. Now it would be very remarkable if any system existing in the real world could be exactly represented by any simple model. However, cunningly chosen parsimonious models often do provide remarkably useful approximations. For example, the law PV = RT relating pressure P, volume V and temperature T of an *ideal* gas via a constant R is not exactly true for any real gas, but it frequently provides a useful approximation and furthermore its structure is informative since it springs from a physical view of the behavior of gas molecules [118]." For such a model there is no need to ask the question "Is the model true?". If "truth" is to be the "whole truth" the answer must be "No". The only question of interest is "Is the model illuminating and useful?" [118]. Box was a famous mathematician and statistician, some of his many of his quotes are still used today.

"Due to the rapid rate of increased in product complexity and need to shorten delivery times, the Model-Based Development (MBD) process has been adopted to help manage the complexity of these systems while making product development more efficient. Adopting MBD has resulted in toolchains that allow for efficient rapid controls prototyping, automatic code generation, and advanced validation and verification techniques, such as Hardware-in-the-Loop (HIL). Requirements traceability is necessary for the MDB process and grows more complex when considering the many artifacts that need to be handled for V&V testing. With the compliance requirements of DO-178C and ISO26262, it is even more critical to tract the development and testing process [90]." While discussing MDB, it is important to consider who is responsible for each step of V&V. Figure 2.18 details the Systems "V" with responsibilities.

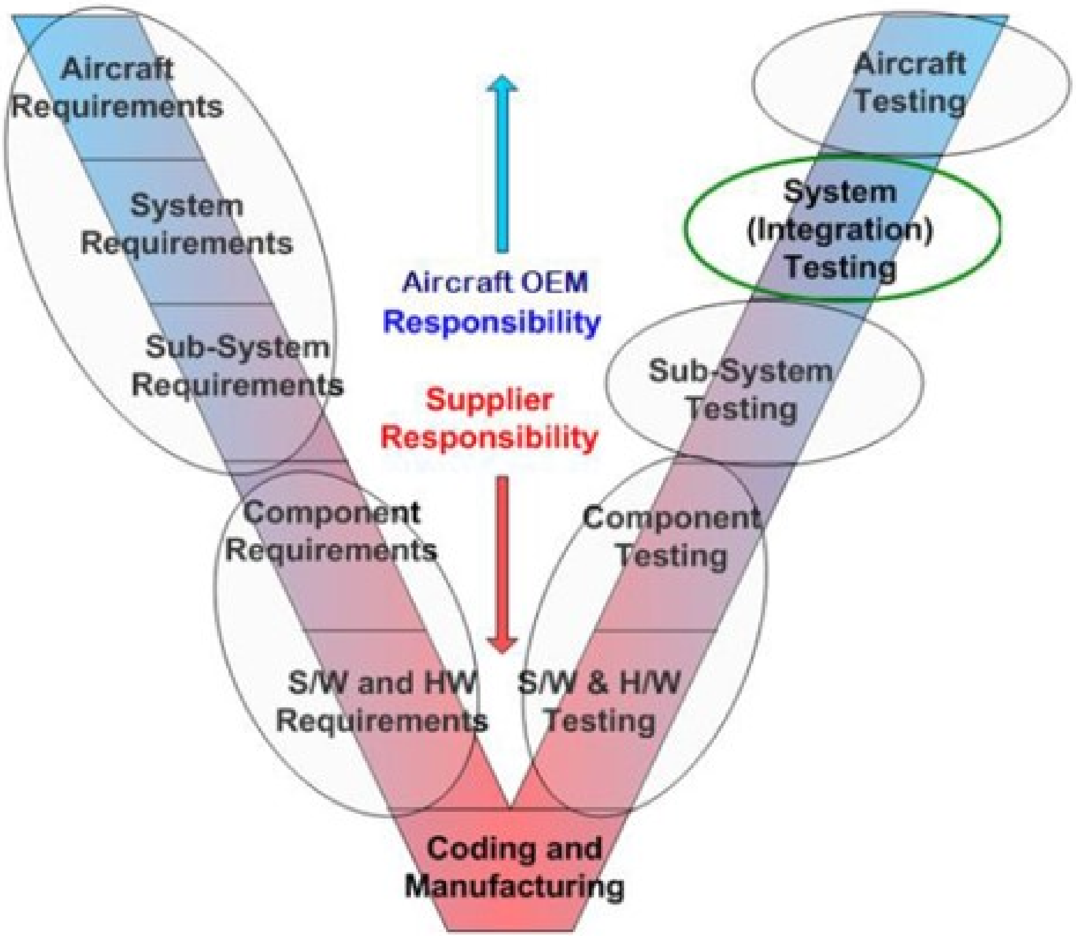

Figure 2.18: V-Cycle Stages for MBD-Based V&V [22]

NASA has studied V&V of controls based on simulations. "With the dramatic growth of model-based control paradigm, tools and methods are needed to demonstrate the complains of this class of control systems with design and safety requirements, also in accordance with specific certification processes, such as the process prescribed by DO-178C. An ongoing project funded by NASA is being carried out to develop advanced techniques of the V&V of model-based control systems. This V&V framework is based on a series of structured steps, first decomposing mission goals into system functional and logic specifications, then applying time-dependent multi-valued logic tools such as DFM and Markov-CCMT and their formal inductive/deductive logic analysis to demonstrate the correctness of system specifications (design validation step) and the correspondence of actual system behavior to such specs (system verification step). The V&V framework also fits within a GSN safety case architecture, whereby safety goals are successfully decomposed into risk scenarios, which can be prioritized using risk informed criteria, and for which design coverage can be shown by means of the evidence provided by the DFM and Markov-CCMT logic analyses [63]." See Figures 2.19 and 2.20.

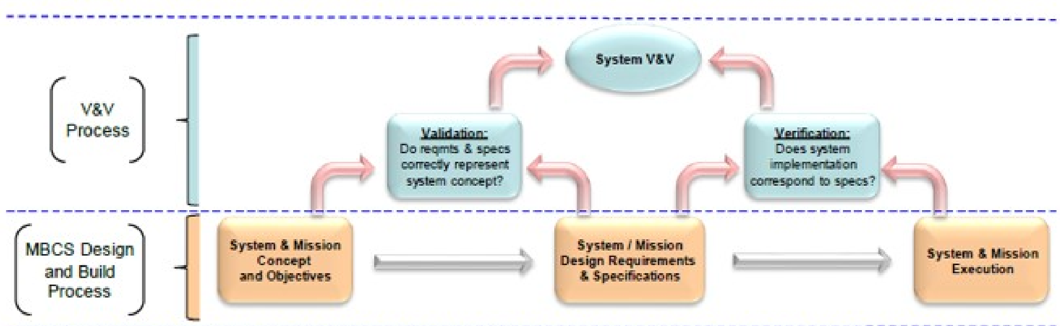

Figure 2.19: Two-Stage V&V Process [63]

When attempting to validate a model, it is necessary to first build a model that is robust enough that it is a nearly accurate representation of the actual en-

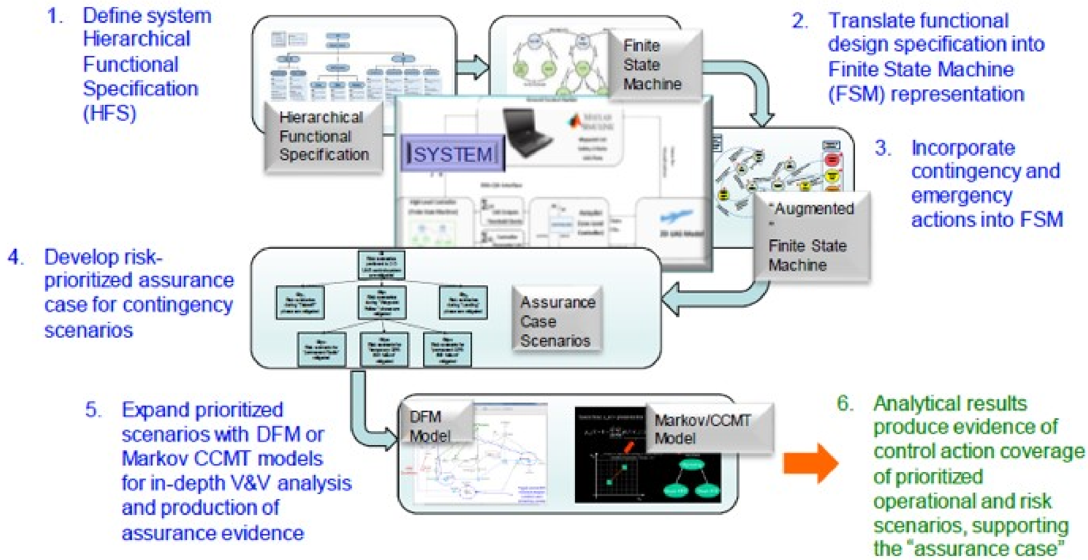

Figure 2.20: MBCS V&V Framework Process Flow and Elements [63]

vironment. Berger and Tischler, along with their team, developed a model of the Calspan variable stability Learjet by stitching together multiple smaller models that consisted of various trim conditions. The final model was validated by comparing its performance against actual flight data [27].

2.7.6 FAA See and Avoid Research for Autonomy

For a vehicle to operate in the national airspace, the FAA requires that they have the ability to detect and avoid other aircraft. The requirements for a UAS to complete this task are more stringent that required by manned aircraft. "To safely avoid another aircraft, an unmanned aircraft must detect the intruder aircraft with ample time and distance to allow the ownership to track the intruder, perform risk assessment, plan an avoidance path, and execute the maneuver [119]." While this definition may seem easy to accomplish by a pilot, the ability to quantify the requirements mathematically, and prove that an unmanned system can fulfill them is a daunting task. Figure 2.21 was developed by Wikle et al. in an attempt to define the see and avoid requirements for UASs to operate in the national airspace system.

2.7.7 Naval See and Avoid Certification

The RQ-21A Blackjack and the RQ-7B Shadow UAS are the first examples of a military UAV to be given access to the national airspace system under extremely limited circumstances. The main issue has been see and avoid. Allowing an aircraft to operate in uncontrolled airspace has always been dependent upon the individual pilots accepting see and avoid responsibilities. This is difficult when there is not a pilot on board. The Blackjack and Shadow UASs have the requirement to fly through uncontrolled airspace after they are launched form MCAS Cheery Point until they are able to reach restricted airspace.

For *see and avoid* NAVAIR 4.1 (Systems Engineering and Technical Support Services) was the flight clearance authority for a ground based sense and avoid system [120]. The idea behind the system was for it to monitor all traffic that could affect the Blackjack or Shadow UASs. Depending on the traffic, it would issue a *Go* or *No Go* for launch. The FAA authorized this system to fulfill the *see and avoid* requirement normally accomplished by a qualified pilot for the limited conditions required for the UASs short flight form MCAS Cherry Point to the restricted area. As of May 2018, this is the only known case where the FAA has allowed a military

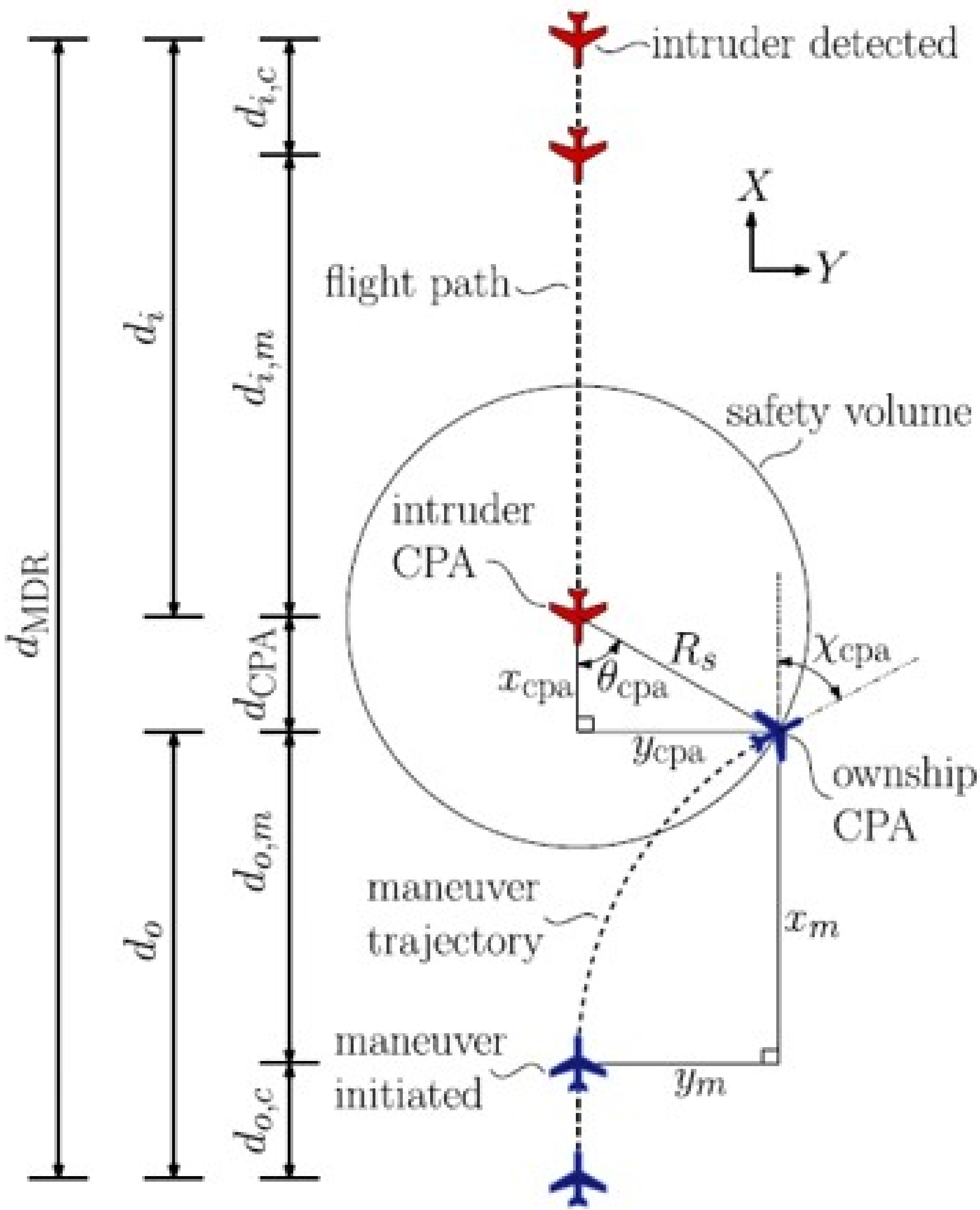

Figure 2.21: The Total Minimum Detection Range, d_{MDR}, Needed and a Representation of the CPA [119]

UAS to operate in the National Airspace System without extensive risk mitigation steps, where technology has been used to accomplish a task normally reserved for a qualified pilot. Of note, while NAVAIR 4.0P certified the Blackjack and Shadow

UASs, NAVAIR 4.1 was the certifying official for the *see and avoid* task. This may lead to an alternative certification path for autonomous functionality in the future.

2.7.8 AACUS

As a possible test bed for our research, NAVAIR has offered partial use of the AACUS autonomy demonstrator as a test bed. The intent was to demonstrate that various levels of autonomy are currently possible.

AAUCS is based on a simple architecture. A number of sensors (visual, LiRAR, and IR) are combined to build SA. multiple computers were added to the cargo area to serve as the decision engine (this is the TALOS decision engine) for an autonomous UH-1. The decision engine then decides where the aircraft will fly and makes inputs to the flight controls to complete the mission.

As a viable method for certifying a naval aircraft to fly without a pilot (or operator in the case of UAVs) in the loop does not exist, the prime contractor used an experimental certificate from the FAA to certify AACUS for fight. The FAA did not have any concern with TALOS as the safety pilot would be responsible for safety of flight. The FAA confirmed that the flight control inputs were within the limits of the aircraft, and mandated a pilot would be at the controls ready to take over at a moments notice (similar to the currently certification of “driver relief modes” for cars such as Teslas).

The use of hardware in the loop is critical for validation of controllers such as TALOS on AACUS. When putting a controller on a known system, it is a necessary to show flight clearance officials that the software will perform as designed when installed. “Hardware-in-Loop Simulations (HILS) are an integral part in the validation of any system under development, more-so, in case of Aerial Vehicles since flight testing of the vehicles is not always possible [29].”

The TALOS system is the decision engine behind the AACUS optionally manned UH-1. The various algorithms which control the actions of flight controls are programmed into it. Hardware in the loop testing was critical to the system moving into the flight test phase. TALOS takes in the various sensor inputs, builds SA on what is happening around it, and manipulates the flight controls to accomplish its assigned mission. It basically performs the roll of a qualified pilot. The purpose if this research is to propose a valid approach to naval flight clearance officials that a decision engine such as TALOS can perform behaviors that are currently reserved for qualified pilots. Chapter 4 covers the AACUS system, and flight test of AACUS, in more detail.

Farinella, Lay and Dhandari examined collision avoidance and path planning for small autonomous UASs. Their research focused on methods to operate autonomously and safety in obstacle rich environments using “Predictive Rapidly Exploring Random Tree (RRT) algorithm to safety navigate around multiple obstacles or other aircraft. The RRT algorithm guarantees a collision-free path, and maneuvers UAS’s around randomly generated dynamic obstacles in a simulated environment to the specified goal waypoint [121].” Their algorithm “assumed the availability of Automatic Dependent Surveillance-Broadcast (ADS-B) sensors and secondary sensors such as scanning LiDAR for collision detection [121].” They showed, in a simulated environment, that having theses sensors coupled with the Predicative RRT algorithm guaranteed collision free autonomous navigation in a dynamic 2D environment [121].

The TALOS/AACUS uses RRT* as a path planning method. The system uses its sensors to build its SA on its surroundings, then uses RRT* to determine the path towards its flight objectives. RRT* is a rapidly exploding random tree algorithm that can generate an optimum path through the tree network. The level of optimization depends on the amount of nodes used its network. The drawback to RRT* is that the optimum path is difficult to define real time as the number of nodes increases. Lee, Lee and Shim developed a receding horizon based RRT* algorithm that limits the number of nodes and enables the near optimum path to be computed real time. They demonstrated this using a six DOF quadrotor model within Simulink and simulated is motion through a maze. "We developed a real-time path planning method based on the RH-RRT* algorithm. In order to overcome the disadvantage of RRT*, for which the computation time sows according to the number of odes, our algorithm continuously performs node removal and updates the biased random sample to a point in the receding horizon area for effective sampling [122]." Figure 2.22 is a graphical presentation of the RRT and RRT* algorithms in use to define a path around obstacles [122].

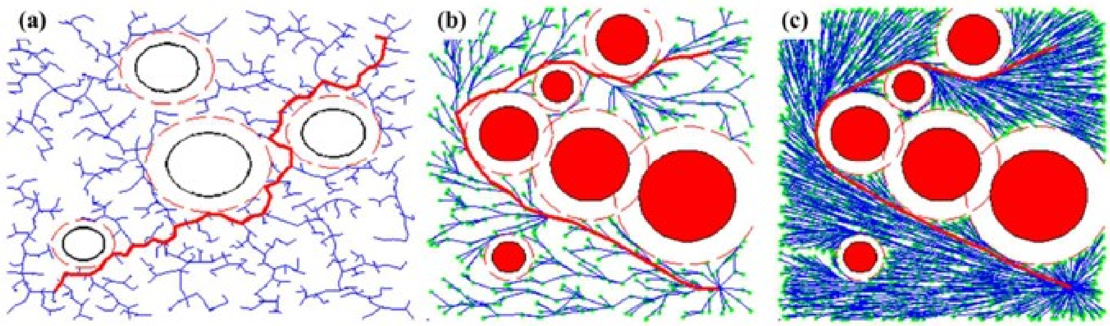

Figure 2.22: RRT and RRT* Path Planning. Image (a) is an Example of RRT, Which has been Widely used for Path Planning of Autonomous Robotic Path Planning. Image(b) is a Midpoint of the RRT* Algorithm Where it is Defining a More Optimal Path Through the Obstacles. Image (c) is the Final Product Where the RRT* Algorithm has Defined the Optimal Path Around the Obstacles [122]

2.8 Helicopter Landing Mission Overview

The landing mission is a difficult regardless of aircraft type. For an autonomous system to select the proper location there are several issues that need to be considered. During flight loss of power, catastrophic system failures and unforeseen circumstances can necessitate an aircraft making a forced landing. This was seen when US Airways Flight 1549 was ditched in the Hutson River after multiple bird strikes caused both engines to fail at an altitude that negated any possibility of reaching a prepared runway [123]. Pilots are trained to constantly be on the lookout for landing locations in case an emergency landing is necessary. In 2016 a Technical Note was published describing this problem for forced lands of current UASs. The Note recommended that the community develop algorithms that can determine the best place to land in case of an emergency [33].

The idea of using laser based sensors as a UAS landing sensor is not new. In 2014 a team from the University of Kansas (Lawrence) conducted a number of experiments to determine in a laser altimeter could be used under varying ground condition (different colors, roughness, climate of landing surface). They pointed put that almost 70% of UAV crashes take place during the landing phase of flight. Having a reliable sensor across a number of variables would be beneficial to the UAS community [124].

When attempting to land a UAS in an unprepared environment there must be some allocation for safety. One of the safety concerns is that there are not any moving objects in the landing zone (such as trucks or tanks). Numerous UAV based vision sensors have solved this problem. One example is the use of synthetic basis (SYA) feature descriptor to perform frame-to-frame feature matching to identify if an object moves from one frame to the next [69].

NASA is preparing to return to the Moon. However, this time they intend to land large vehicles that may not be manned. During the Apollo missions, the Commander had the ability to control the descent, and pick a safe landing site. If next generation of lunar delivery vehicles are to be autonomous, they would need a way to choose a safe landing site. In the spring of 2014, NASA demonstrated a "Guidance, Navigation, and Control (GN&C) and LiDAR-based sensing system autonomously scanned a lunar-like hazard field from an autonomous, rocket-propelled, free-flying lander on a lunar-like approach trajectory, then correctly identified a safe site, and subsequently provided closed-loop precision guidance for landing on that safe site [70]." The AACUS system uses LiDAR to find hazards and TALOS to determine a safe landing zone through a similar approach.

2.8.1 Confined Area Landing (CAL)/Landing Zone (LZ)

Every naval aviator (pilot) utalizes a NAVAIR approved pilot's checklist when they fly. Helicopter pilots are no exception. The SH-60 community uses the A1-H60RA-NFM-500 checklist [125]. This pocket checklist (PCL) contains emergency procedures, normal procedures and briefing materials. Prior to attempting landing at a CAL/LZ the PCL contains a number of items the crew needs to brief for safety. They include:

- Location (MGRS/lat-long): Helps to properly identify the CAL/LZ and be input into internal systems.

- Depiction (chart/drawing/photo): Helps the crew prepare for what they can expect to see when they reach the unprepared CAL/LZ.

- Site Evaluation: Allows the crew to determine if the location is suitable and what the hazards they may expect to find once they arrive at the location.

- Orientation: Magnetic Heading: On which heading will be optional for the approach to the CAL/LZ

  Landing Point: After studying the site, the crew can determine the optimum landing site.

- Markers (panels/smoke): What visual cues can the crew use to determine low level winds (vital for helicopter operations).

- Waveoff Procedures:

Waveoff Criteria: What could happen that would necessitate a waveoff

General Heading: What is the optional path for a waveoff, power available and obstacle avoidance are consideration in this decision

Obstacles: As the aircraft starts descending into a CAL/LZ it is likely that he pilot will not have visual on obstacles, he relies on this crew chief(s) to visually clear the helicopter and relay the current status verbally to the pilot. This verbal que helps the pilot maintain SA on the approach.

Effects of wind/dust/snow/debris: These effects can have a dramatic effect on the safety of flight of the vehicle it is critical that the crew brief the contingencies.

Reentry Procedures: If a waveoff is executed, a reentry procedure needs to be discussed.

2.8.2 SWEEP

Landing in an unprepared LZ is a difficult mission for qualified HACs. The last 15 years has seen several fatal mishaps where naval aviators have made decisions that lead to unsuccessful landing attempts. The CNAF, established a procedure for pilots to complete when attempting a landing in such a location. The procedure is abbreviated as SWEEP (Size/Slope, Wind, Elevation, Escape Route, Power) [125]. Sweep is also detailed in Section 3.2.1 and 4.1):

- **Size**: The S in SWEEP has two meanings, the first is size of the LZ. The HAC must be able to define the size of the LZ from altitude (nominally 200 ft Above

Ground Level (AGL)). This includes obstacles and actual area and orientation available for the vehicle to touch down in. An obstacle within the LZ may not negate the suitability of the LZ. Rotor wash may blow some items out of the way during landing (such as tumbleweeds). A HAC uses their experience and judgment to identify which objects may pose a threat. West coast helicopter pilots normally train in the desert of eastern San Diego. The biggest threat to defining a LZ are tall bushes that can cause the vehicle to tip over if they are under the aircraft on landing. A confined area, such as an urban setting, offer still other issues dealing with the actual dimensions of the LZ. Buildings and fences confine the available space to land in. HACs are expected to be able to visually identify the LZ and determine the suitability for landing. All helicopters differ in size.

- **Slope**: The S in SWEEP also stands for Slope. Most prepared LZs are flat and clear of any obstacle. When a helicopter touches down on a flat surface both skids, or landing gear, touchdown at nearly the same time. The greater the slope the more of a risk the vehicle may tip over on landing/touchdown due to dynamic rollover. The risk comes when only one of the two main touchdown points makes contact with a surface and becomes a pivot point for the vehicle. Standard operating procedures list a limit for slope based on vehicle configuration and environmental conditions. HACs are expected to evaluate the slope for suitability from altitude, and continually evaluate the LZ through touchdown.

- **Wind**: The W in SWEEP stands for wind. Unlike their fixed wing counterparts, helicopters normally do not land with a forward velocity that dominates the local wind during landing. A fixed wing aircraft may be able to withstand crosswinds of 30+ kts due to its forward velocity of 100+ kts. A helicopter may have crosswind limits of 5-10 kts while landing. A HAC is expected to evaluate the landing area before approach and continuously during approach to ensure the aircraft can complete a safe landing. In a CAL/LZ, when an aircraft gets near the ground the wind has a tendency to shift greatly due to local conditions. These shifts may be difficult for the HAC to anticipate from altitude. The HAC is expected to abort a landing if an unsafe wind condition is present.

- **Elevation**: The first E in SWEEP stands for elevation. Tactical helicopters are historically under powered due to their weight. The closer to sea level the better the performance of the engines on the aircraft. As altitude increases the performance of the engines is reduced. The USN trains selected naval aviators at the mountain training school in Fallon, Nevada. There pilots learn how to control their aircraft when its performance is limited due to elevation. A HAC is expected to be able to accurately evaluate the vehicles performance based on the altitude of the LZ. They are also expected to abort the landing if an unsafe condition exists.

- **Escape Route**: The second E in SWEEP stands for escape route. When evaluating an unprepared LZ, HACs are expected to be able to find a way out (if one exists). The way out is used as an escape route when aborting a landing/approach. This route may be used when an unexpected unsafe condition develops. One example would be if the LZ becomes fouled by an interloper (such as a moving vehicle or wild life). On this step of the SWEEP procedure the HAC must select their escape route if a safe landing can no longer be executed. If any escape route does not exist, some low priority missions will be aborted as the extra risk associated with the mission is not acceptable based on the priority level.

- **Power**: The P in SWEEP stands for power. As with all aspects of vertical lift aviation, power is the most critical part of aircraft performance. The two main expressions are HIGE (Hover In Ground Effect) and HOGE (Hover Out of Ground Effect). These values define the power margin available to the pilot on the day in question and are constantly evaluated during flight as conditions change. Environmental factors, such as temperature and density altitude, combined with mechanical factors (the actual performance of the engines installed on the vehicle), define the power available to the pilot for use. A HAC is expected to be able to evaluate the power they have available for approach to determine suitability.

2.9 Helicopter Aircraft Command (HAC) Qualification

The purpose of this research is to determine a path forward for certifying a decision engine to act as a qualified pilot. For the helicopter community, this equates to being designated as a HAC. To accomplish this, the current HAC qualification process must be understood. Following graduation from the Helicopter Replacement Air Group (RAG) a pilot will be assigned to a fleet squadron for approximately 36 months. During this time they will be expected to qualify as a second pilot, complete a HAC syllabus, complete the prerequisite flight experience in model, pass a HAC oral board, and ultimately earn their COs trust in their decision making process before they are considered a fully qualified HAC.

2.9.1 Helicopter Second Pilot

Prior to being designated a HAC, a pilot must complete demonstrate proficiency in a number of areas relating to their aircraft. The following are excerpt from CNAF M-3710.7 [10].

Helicopter Second Pilot: In addition to being a designated helicopter pilot, a helicopter second pilot shall:

A: Have pilot hours in class and model as required by the command officer or higher authority and demonstrate satisfactory proficiency in the following:

- Ground Handling
- Flight technique in normal and emergency procedures for flight including autorotation and the use of flotation gear, if applicable
- Navigation (all types applicable to unit mission and model aircraft)
- Tactical employment of the aircraft and associated equipment in all tasks of

the unit mission

- Night tactical operations and operational instrument flying within the capability of the model

B: Possess a current instrument rating

C: Demonstrate knowledge through oral and/or written examination on the following

- Model aircraft and all associated equipment.
- Operational performance in all flight maneuvers.
- Weight and balance.
- Appropriate NATOPS manual.
- Survival and first-aid.
- Applicable technical orders and notes, OPNAV instruction, FAR, ICAO procedures, SCATANA plans, and NAVAIRSYSCOM instructions and technical directives.
- Search and rescue procedures.
- Communication
- Unit mission and tactics
- Navigation.

- Flight planning.
- Local and area flight rules.
- Fleet and type tactical instructions and doctrine.
- Applicable portions of NWPs, FXPs, JANAPs, ACPs, and ATPs.
- Recognition applicable to unit missions.

D: Satisfactorily complete a NATOPS evaluation in model

2.9.2 HAC Syllabus

Prior to sitting their HAC board, a HAC candidate is expected to complete a number of syllabus events. These events range from simple navigation flights, to complicated training flights detailed by the Air Combat Weapons and Tactics Syllabus. Like their tactical jet counterparts, naval helicopter pilots are expected to complete a number of tactical events prior to being authorized to serve as an aircraft commander (HAC).

2.9.3 HAC Requirements

CNAF M-3710.7 states [10]: Requirements listed below are to be met by pilots qualifying in multiplied rotary-wing aircraft. COs are qualifying authorities, or higher authority, shall prescribe proficiency standards, detailed factors, and specific minimums based on this chapter, class and model aircraft, and the unit mission. Within each classification, the weight and emphasis on the factors enumerated must be determine by the activity. Waivers of minimums may be granted by the appropriate immediate superior in command commensurate with demonstrated ability and only when deemed necessary to accomplishment of the unit mission.

To be qualified as a helicopter aircraft commander, the NATOPS manual shall establish the designation for the particular model, and an individual shall:

- Have completed the requirements for and possess to an advanced degree the knowledge, proficiency, and capabilities of a second pilot.
- Have a minimum of 500 total flight hours.
- Have 150 flight hours in rotary-wing aircraft.
- Have pilot hours in class and model required by the CO or higher authority and demonstrate the proficiency and judgment required to ensure the successful accomplishment of all tasks of the unit mission.
- Demonstrate ability to command and train the officers and enlisted members of the flight crew.
- Demonstrate the qualities of leadership required to conduct advanced base or detached unit operations as officer in charge when such duty is required as part of the units mission or method of operation.

2.9.4 HAC Oral Board

The naval HAC oral board can vary drastically depending on the individual squadron (as squadron leadership changes every 15 months), and the squadrons mission is not consistent across the USN. The primary researcher had the opportunity to sit on three HAC boards as an observer to get a better understanding of their composition and goals. The boards were for candidates from two different squadrons: HSC-14 and HSM-73.

The various boards were similar in nature, the XO (second in command) was typically the senior member. With the exception of one junior officer (normally a LT, O-3, who serves as the squadron NATOPS or Pilot Training Officer) the other 4-5 members were field grade officers (LCDR, O-4) and heads of departments in the squadron (Safety, Maintenance, Operations, Tactics). The typical rank of the HAC candidate was O-2 (LTJG). The idea of the board membership being significantly senior to the candidate is designed to put the candidate under stress. A typical board length was two hours.

The junior officer on the board was normally the NATOPS or Pilot Training officer. Their questions were geared to test the candidate's basic knowledge of the limitations and standards of operations of the helicopter. The answers require rote memory, and no critical thinking. A sample question would be "what is the oil pressure limitations at max continuous N2?" Or, "What is the required number of rescue swimmers required for overwater SAR?"

The department head board (field grade officers) members questions were all geared toward scenario based questions. They were designed to test the candidate's critical thinking in situations they may be placed in once they are a qualified HAC. The scenarios were varied depending on the personal experience of the board member and the primary mission of the squadron. For the HSC squadron, the questions were geared more towards logistics and SAR. The HSM squadron boards tended to focus on mission critical decisions. In both cases, the senior board member needed to be convinced the HAC candidate had a grasp of the situation, the capabilities of their aircraft, and they had the ability to think outside of the box. Providing the senior member was confident in the candidate's performance, they would recommend the HAC qualification to the CO (who has the ultimate decision to qualify the candidate as a HAC).

2.9.5 Commanders Trust

To qualify a candidate as a HAC, the CO is placing trust in the pilots' judgment. Any pilot can follow directions, or complete a simple mission when everything goes as planned. The question is how they will respond when things don't go as planned. By designating a pilot as a HAC to CO is putting their stamp of approval on the pilots ability to cope with the unexpected. The CO of HSM-71 had an interesting scenario for HAC candidates. It places the HAC with is a situation where there is not right answer. He gives a scenario where the HAC is asked to perform a one way mission, with no guarantee of safe recovery at the end.

The COs and XOs of HSC-14 and HSC-73 were intrigued by the possibility of certifying an unmanned helicopter. They felt that it was possible to program a vehicle to perform simple tasks, but were hesitant in believing it could perform as a fleet qualified HAC under unplanned situations. They agreed that it is the future, but were glad they would not be tasked with certifying a decision engine to act as the HAC in their respective squadrons at the time of our interview.

Chapter 3: Requirements Definition

The last 15 years has seen a large uptick in the use of unmanned aircraft. However, current Safety of Flight (SOF) clearances for unmanned aircraft require a qualified operator who can make decisions and ultimately bears the responsibly for the safe operations of the vehicle. The future of aviation is unmanned, and ultimately autonomous. Yet, a clear path for certifying an autonomous vehicle to make decisions currently reserved for qualified pilots does not exist. This chapter presents a preliminary approach for certifying an autonomous controller to select an appropriate landing site for a large rotorcraft in an unprepared landing zone, and focuses on the first four steps of the methodology proposed in Section 1.4.

In an attempt to provide a path forward for certifying autonomy in aviation, this chapter provides a limited approach for providing evidence that can be used for certifying an autonomous controller to exhibit non-deterministic behavior when selecting a LZ autonomously during the unprepared CAL/LZ mission. This mission (the task of selecting and continuously evaluating a landing spot during the approach and landing phase of flight) is currently carried out by the USN and USMC helicopter communities [125]. Prior to certification, TAEs need to be provided certification evidence that the system can complete tasks currently reserved for pilots [9]. This chapter will decompose the tasks currently completed by a pilot during the CAL/LZ mission to their basic requirements. To develop these requirements, we consulted (over several interview sessions) multiple senior naval officers (those that currently certify a pilot as a Helicopter Aircraft Commander (HAC)), and followed several junior aviators during the qualification process. Through our conversations and observations we gained insight as to what was expected of a fully qualified HAC during the mission. Ultimately we propose a clearance envelope where the system can exhibit non-deterministic behavior. This means the actions of the system cannot be exactly predicted by evaluating the systems parameters, and the system is clear to make decisions currently reserved for qualified pilots providing it does not reach one of the limits of the clearance envelope. For the CAL/LZ mission, this implies that the autonomous controller can pick its landing spot providing it does not violate restrictions put in place. If the system were to reach one of these limits, it would revert to pre-determined behavior. We examine the correctness of the specification in an effort to show that a path forward exists in which formal verification could be used to certify autonomous systems to complete tasks currently reserved for qualified pilots [36]. We used Prototype Verification System (PVS) (a theorem proving tool) to examine a high level specification for correctness. Then the analyzed specification was used to develop a protocol for the actions the autonomous controller would take when selecting, and controlling the aircraft during the CAL/LZ mission. The protocol was then evaluated against a sample set of possible LZ conditions to ensure that only a LZ that met all of the requirements of the specification would be allowed to be selected by the autonomous controller (eliminate corner cases).

Software developers can use the protocol as a guideline for developing the specific code that will control the aircraft. We also presented the protocol to the same senior naval officers that helped develop the requirements for the specification to ensure that it met their criteria for qualification of HACs. All four naval officers agreed that, provided the assumptions were valid, the protocol was adequate for modeling the behavior of a fully qualified HAC in the CAL/LZ mission. The evaluation can be used by certification officials as evidence for the ultimate certification of the system [9].

This chapter is structured as follows. Section 3.1 is a summary of the qualification process for naval aircraft and naval aircrew (more detailed information is available in section 2.3). Section 3.2 defines the requirements a decision engine would be required to complete when completing an unprepared landing and develops a specification to meet those requirements. Section 3.3 provided a analysis of the specification to demonstrate how a process can be used to show the specification meets the requirements of the system. Section 3.4 proposes a protocol that software designers can use when developing the control laws of the autonomous vehicle. Section 3.5 summaries the chapter.

The contributions of this chapter include:

- Definition of the requirement a decision engine must complete if it were to be approved to complete the CAL/LZ mission autonomously (a task currently reserved for a qualified pilot).

- Development of a state machine specification which follows the various states required for an autonomous vehicle to complete the CAL/LZ mission.

- Analysis of the above specification to ensure it meets the requirements for a decision engine to be certified to complete a task currently reserved for qualified pilots (CAL/LZ mission).
- Development of a protocol that software designers can use for programming a decision engine to complete a task currently reserved for qualified pilots (CAL/LZ mission).

3.1 Naval Aviation Certification Processes

3.1.1 Current Certification Process for Naval Aircraft/Systems

Currently, when an aircraft is certified safe for flight (when operated safely, they will not break down or cause a danger to the general public) it is assumed that they will be operated by a qualified pilot (or operator in the case of large UAVs such as Global Hawk or Predator). As an example of a currently fielded system, the USN currently operates the MQ-8 Fire Scout UAV. NAVAIR has certified the large rotorcraft to fly without a qualified HAC on board. However, an Air Vehicle Operator (AVO) is ultimately responsible for the safe operation of the vehicle. During pre-flight mission planning the AVO programs the vehicle to complete parts of the mission without operator input (similar to an autopilot). In the event of loss link, the system will fly to a pre-planned point, and land. The system does not perform any evaluation of the landing point, it simply executes a pre-planned route to a LZ and auto-lands [126].

NAVAIR 4.0P has established processes where TAEs, who have been given the authority in their subject fields, review relevant artifacts prior to approving their portion of a flight clearance. Artifacts can range from SME opinion to detailed engineering analysis. Often an artifact is a data set characterizing the performance of a system. In the end, artifacts exist to quantify the system and allow the certification official to determine the risk they will be accepting.

For the respective TAEs to certify autonomy in their subject area, several challenges will need to be overcome. In the words of the former chief engineer of the USAF: "It is possible to develop systems having high levels of autonomy, but it is the lack of suitable V&V methods that prevents all but relatively low levels of autonomy from being certified for use [93]." The AFRL funded a study asking a question regarding the state of possible processes for certification of UASs which employ machine learning or autonomous functionality through some sort of evidence based licensure process. The report summarized several categories that may lead to the certification of UASs. These categories were:

- Formal Methods
- Requirements and Metrics
- Normative Oracle Generation
- CoActive Design
- Implications of Learning Autonomous Systems

- M&S Considerations for Licensure of Autonomous systems [94]

All or some of these categories will be required for the individual TAEs to accept the risk associated with certifying the autonomous functionality.

3.1.2 Current Certification Process for Helicopter Aircraft Commander (HAC)

The overarching purpose of the research presented in this chapter is to determine a path forward for certifying a decision engine to act as a HAC in the USN or USMC. To accomplish this, the current HAC qualification process must be understood. This process is formally established, but full qualification depends on a subjective decision of a CO (typically an O-5 or O-6) [10]. Following graduation from the helicopter RAG a pilot will be assigned to a fleet squadron for approximately 36 months. During this time they will be expected to qualify as a second pilot, complete a HAC syllabus, complete the prerequisite flight experience in model (such as a H-60 or H-1), pass a HAC oral board, and ultimately earn their CO's trust in their decision making process before they are considered a fully qualified HAC [10].

To qualify a candidate as a HAC, the CO is placing trust in the pilot's judgment. Any pilot can follow directions, or complete a simple mission when everything goes as planned. The question is how they will respond when things do not go as planned. By designating a pilot as a HAC the CO is putting their stamp of approval on the pilot's ability to cope with the unexpected.

3.2 Requirements Definition and The Specification

Prior to SOF certification, officials require data to justify such a flight clearance [9]. This data is referred to as certification evidence. This chapter describes the development of certification evidence for SOF certification of a well-defined task: Autonomous landing of a helicopter in an unprepared landing zone. An unprepared landing zone is a location that is not certified for rotorcraft operations (not an aerodrome or helipad). We use the unprepared Confined Area Landing/Landing Zone (CAL/LZ) mission currently carried out by USN and USMC helicopters communities as a running example [125]. This mission can be as simple as landing in an open field adjacent to a highway, or as difficult as landing between buildings in an urban setting. The process for choosing a landing spot is complicated, and prior to being certified as a HAC a candidate is expected to be able to accurately complete this task [10].

Since the dawn of aviation, many of the innovations we currently take for granted came from the military (some examples include: radar [78], medevac air ambulance [79], jet engines [80], glow sticks [81], and advanced night vision technology [82]). Many military applications can transition easily to the civilian sector, as their functionality is similar. For this reason, we chose a military application that can be easily translated into a civilian sector for this research. The evidence generated can be use for certification of future autonomous vehicles.

For naval aviation, airworthiness certification authority is delegated to Naval Air Systems Command (NAVAIR) 4.0 Engineering (4.0P is the branch assigned)

[9]. When a new capability (i.e., software, weapon or air frame) is acquired, and before naval personnel operate it, 4.0P must grant a flight clearance (also referred to as a SOF certification). The certification process for naval aircraft is a risk mitigation process. Aircraft subsystems, software, components and ultimately the aircraft itself are certified through an established process. Technical Area Experts (TAEs) are tasked with reviewing certification evidence (referred to as artifacts) in their individual technical areas. These reviews are rolled up in to a larger flight clearance which certification officials uses to certify the vehicle as a whole. When a vehicle is certified safe for flight, NAVAIR 4.0P is certifying that when given to a qualified pilot they can safely complete the desired mission of the aircraft [9].

3.2.1 Development of the Basic Requirements

The first step in a path for a flight clearance of an autonomous system to complete tasks currently reserved for a qualified pilot is to define the requirements the decision engine must complete. Landing in an unprepared LZ is a difficult mission for qualified HACs. The last 15 years has seen several fatal mishaps where naval aviators have made decisions that lead to unsuccessful landing attempts. The Chief of Naval Air Forces (CNAF), established a procedure for pilots to complete when attempting a landing in such a location. The procedure is abbreviated as SWEEP (Size/Slope, Wind, Elevation, Escape Route, Power) [125]. Several syllabus flights are dedicated to mastering this task, and these flights must be passed before a pilot can be designated a HAC. These flights consist of 17 events totaling 36 flight hours. The experience gained by completing the syllabus events, in addition to the experience the HAC candidate obtains during other events, is used to train the judgment of the aviator prior to their CO designating them as a HAC [10].

If a decision engine were to be allowed to make the decision on where to land, it would need to demonstrate the ability to complete the SWEEP procedure. This work attempts to translate the judgment used to complete the SWEEP checklist into a decision engine, then allow the decision engine to select a landing point (provided SWEEP is valid). Any protocol used to control its action must prove that it can accurately complete the procedure, every time, before it is certified. It is important to understand each part of SWEEP as detailed in Section 2.8.2.

This chapter proposes a clearance envelope where the decision engine can exhibit non-deterministic behavior. If the vehicle reaches one of the edges it would abort the approach and proceed to a predetermined point. The question is how to define the edges. Using SWEEP as an outline, a protocol can be developed based on a specification for keeping a vehicle within the clearance envelope. We then systematically examined the specification in an effort to ensure it satisfies the requirements defined above. This will serve as an artifact for flight clearance officials to accept the risk of allowing a decision engine to make a decision (landing) normally reserved for a qualified HAC.

3.2.2 Specification

For the limited purpose of defining a specification for the landing of a large rotorcraft in a CAL/LZ using guidance and control from an onboard decision engine, we elected to use a state machine specification [127] (Figure 3.1). The state machine specification follows the various states required for the vehicle to transition from the initial (or reset) point and being safe on deck. Table 3.1 details the various events which happen as the specification transfers from state to another.

The transition states can be summarized as follows:

- **A. "Initial/Reset" State**: At this point the decision engine is at the start of the loop. Following a fuel check (to determine if the current state is above a pre-determined bingo fuel (fuel required to return to a safe landing field)) it will begin the process of selecting a LZ and evaluating it against the SWEEP checklist. If the vehicle is below the pre-determined bingo fuel the decision engine reverts to the "Return to Base" ("RTB") state, and returns to base for more fuel before it attempts the find a valid LZ.

- **B. "Conduct SWEEP Checks to Determine if Selected LZ is a Valid LZ" State**: In this state, the decision engine selects a possible LZ and evaluates the SWEEP checks. If the selected LZ has a valid SWEEP check, the decision engine can then proceed to state C ("Build Ingress Route"). If not, the decision engine retrogrades to state A ("Initial/Reset").

- **C. "Build Ingress Route" State**: In this state, the decision engine builds an ingress route from the start point to a HOGE point. Providing it can be completed with the remaining fuel onboard, avoid obstructions/traffic, and remain within the performance envelope of the vehicle, the ingress route is considered valid and the decision engine can proceed to state D ("Monitor Ingress"). If not, the decision engine retrogrades to state A ("Initial/Reset").

- **D. "Monitor Ingress" State**. In this state, the decision engine monitors the LZ and the performance parameters of the vehicle to ensure that SWEEP remains valid while the vehicle is transitioning from the start point to the HOGE point. Once the vehicle reaches the HOGE point, the decision engine shifts to state E ("HOGE Over Spot to LZ Transition"). If SWEEP were to become invalid prior to the vehicle reaching the HOGE point, the vehicle would execute the escape route, return to the initial/reset point and retrogrades to state A ("Initial/Reset").

- **E. "HOGE Over Spot to LZ Transition" State**. In this state, the decision engine monitors the LZ and the performance parameters of the vehicle to ensure that SWEEP remains valid from HOGE to touchdown. If SWEEP remains valid, the vehicle will complete the mission (land safely). If SWEEP were to become invalid prior to touchdown, the vehicle would execute the escape route, return to the initial/reset point and retrograde to state A ("Initial/Reset").

This state machine specification can be considered a top level. Each of the events described in Table 3.1 have conditions and assumptions built into them. Some

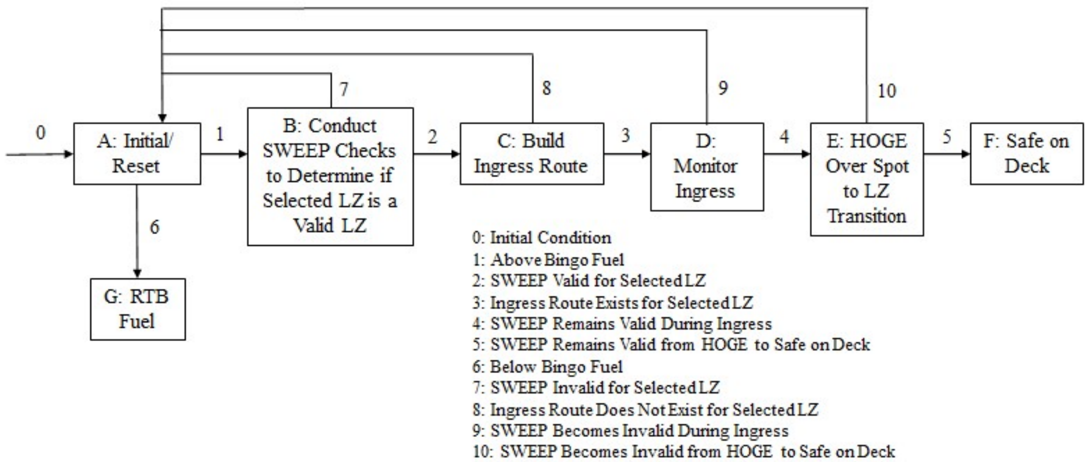

Figure 3.1: State Machine Specification Which Details the Decision Process for a Unmanned System to Make a Decision Currently Reserved for a Qualified Pilot

ID	From State	Events	To State
1	A	Above Bingo Fuel	B
6	A	Below Bingo Fuel	G
2	B	SWEEP Valid for LZ	C
3	C	Ingress Route Exists for Selected LZ	D
4	D	SWEEP Remains Valid During Ingress	E
5	E	SWEEP Remains Valid from HOGE to Safe on Deck	F
7	B	SWEEP Invalid for Selected LZ	A
8	C	Ingress Route Does Not Exist For Selected LZ	A
9	D	SWEEP Becomes Invalid During Ingresss	A
10	E	SWEEP Becomes Invalid from HOGE to Safe on Deck	A

Table 3.1: Event Description for the State Machine Specification Which Details the Decision Process for a Unmanned System to Make a Decision Currently Reserved for a Qualified Pilot

examples of the assumptions are the environmental conditions (weather, atmospheric conditions) and vehicle limitations (actual limits of the air vehicle). These conditions and assumptions must be valid for Figure 3.1 to be a valid flight clearance artifact. Top level assumptions become lower level requirements.

As the specification in Figure 3.1 represents a subset of the overall functionality of the aircraft it has one defined start point ("Initial/Reset" state). From there the decision engine executes the evaluation of possible landing locations until it either completes a safe landing ("Safe on Deck" state) or is forced to abandon the task due to fuel constraints ("RTB" state).

3.3 Analysis of the Specification

In this section we begin with the state machine specification as it relates to controlling the unmanned system in its decision process. We show consistency and completeness via an operational procedure table. We then break down the various processes within the specification into propositions that must be held valid for the specification to be valid. The propositions will then be tracked and analyzed by a theory proving software package to complete the analysis of the specification detailing the decision process for a unmanned system to make a decision currently reserved for a qualified pilot.

Formal methods has been used for aircraft software verification and ultimately certification of aerospace software [36]. The power of formal methods lies in providing precise and unambiguous descriptions and mechanisms that facilitate the development of safety-critical systems in a more robust fashion [128]. By first developing a specification that tracks the various states for landing, then completing the formal methods activities (analyze specification for consistency/completeness, prove the behavior will satisfy the requirements (with assumptions), prove that a more detailed design implements a more abstract one [129]), TAEs can use the results as artifacts for certifying an autonomous controller to complete the CAL/LZ mission. The analysis in this section uses PVS, a theorem proving tool, to examine a high-level specification for an autonomous system in an attempt to certify that the system can complete tasks currently reserved for qualified pilots. This analysis is not a formal verification of the software, but rather a preliminary example of a path toward formal verification of such systems.

3.3.1 Operational Procedure Table

An Operational Procedure Table was used to begin the analysis of the specification (Figure 3.2). The variables along the top row represent the requirements for each associated landing segment (of flight) task (left column) required for the CAL/LZ mission. Each variable has its own assumptions (which would translate to requirements at lower levels). Each task is performed sequentially (top to bottom). Each variable is unknown until the associated segment is complete (changing the variable to a 1 or a 0). A common underlying assumption for all the variables is that the situational awareness provided by the vehicle's sensors to the decision engine is adequate for the current conditions (not degraded to an unsatisfactory level by weather or malfunction).

The following are the variables and their underlying assumptions:

- **Above Bingo Fuel**: The vehicle is above the amount of fuel required to return to a safe landing area. Assumes the fuel management system is functioning properly and the decision engine is able to accurately measure the value.

- **Suitable LZ (Size/Slope)**: The decision engine is able to choose a LZ that is suitable for the vehicle. Assumes the LZ requirements are programmed

LANDING SEGMENT TASK	VARIABLE								
		Initial SWEEP Checks for LZ							
	Above Bingo Fuel	Suitable LZ (Size/Slope)	Winds Within Limits	Valid Elevation Data	Valid Escape Route	Favorable Power Margin	Valid Ingress Route	SWEEP Valid on Ingress to HOGE Point	SWEEP Valid HOGE to Land
Initial/Reset	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
RTB	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
(Return to Hold/Start if a 0 exists in the following		*	*	*	*	*	*	*	*
Conduct SWEEP Check to Determine if Selected LZ is a Valid LZ									
- Size/Slope	1	1	U	U	U	U	U	U	U
- Wind	1	1	1	U	U	U	U	U	U
- Elevation	1	1	1	1	U	U	U	U	U
- Escape Route	1	1	1	1	1	U	U	U	U
- Power Margin	1	1	1	1	1	1	U	U	U
Build Ingress Route	1	1	1	1	1	1	1	U	U
Monitor Ingress	1	1	1	1	1	1	1	1	U
HOGE Over Sopt to LZ Transition (Safe on Deck)	1	1	1	1	1	1	1	1	1

* Return to Initial/Reset Segment if prior to ingress, execute Escape Route to Start Point and then Initial/Reset Segment, Clear Variables

Possible Variable Values

1	Positive
0	Negative
U	Has Not Reached Evaluation Yet
N/A	Not Applicable

Figure 3.2: Operational Procedure Table Converting the State Machine Specification Into the Various Tasks Required for an Unmanned System to Make a Decision Currently Reserved for a Qualified Pilot

properly (size, and slope) and can properly classify obstructions as threat or no threat.

- **Winds Within Limits**: The decision engine is able to compare the current wind conditions to the programmed limits for the vehicle. Assumes the wind limits are programmed properly (head and cross wind).

- **Valid Elevation Data**: The decision engine is able to determine its current Mean Sea Level (MSL) altitude from its internal systems (some combination of Global Positioning System (GPS), Inertial Navigation System (INS) and internal pitot static system).

- **Valid Escape Route**: The decision engine has developed an escape route which will return the vehicle to the start point and remain within safety limits. Assumes the safety limits are developed and defined within the programming of the decision engine.

- **Favorable Power Margin**: The decision engine has defined the power margin (power required/power available) to be adequate for the LZ. Assumes the margin has been defined and programmed into the decision engine.

- **Valid Ingress Route**: The decision engine is able to build an ingress route which will keep the vehicle free from collision and within the flight limits of the vehicle. Assumes the limits of the vehicle are programmed into the decision engine.

- **SWEEP Valid on Ingress to HOGE Point**: The decision engine is able to continuously monitor the LZ during the approach to its HOGE point. Should the status of SWEEP change to invalid, the vehicle would need to abort the approach, execute the escape route, and transition to the reset point.

- **SWEEP Valid from HOGE to Land**: The decision engine is able to continuously monitor the LZ during its landing through touchdown. Should the status of SWEEP change to invalid, the vehicle would need to abort the landing, execute the escape route, and transition to the reset point.

3.3.2 Consistency and Completeness

The operational procedure table (which contains cell values (1, 0, U or N/A) of each requirement) was used to help define consistency and completeness. The table shows consistency by the fact that no two columns are operational for any combination of values for the variables as no two columns have the same cell values (at most one outcome assigned under each possible scenario). The table shows completeness by the fact that for all values of variables only one column is operational as all possible combinations of the variables are listed within the table, and no two columns are equal (some outcome assigned to every possible scenario) [130].

3.3.3 Theorem Proving Model

To prove that the system will complete the task, and show what the system will not do, the top level requirements outlined in Figure 3.2 were separated into three propositions (each of which having supporting propositions (e.g. Proposition 1.1 and Proposition 1.2 and Proposition 1.3 imply Proposition 1.0 is true)) which must remain true for the overall model of a successful landing to be valid. These propositions alone would not satisfy formally verifying the specification. That would require detailed formal analysis of the specification. This analysis would include validating all of the assumptions underneath the top level specification presented in this research. Which in turn would require more explicit definitions than the booleans presented and is beyond the scope of this research.

Proposition 1.0: The LZ is suitable for landing (all of the supporting propositions are true).

Proposition 1.1: The size of the LZ is adequate for the vehicle.

Proposition 1.2: The slope of the LZ is adequate for the vehicle.

Proposition 1.3: The LZ is clear of obstructions.

Proposition 2.0: The conditions for landing are suitable (all of the supporting propositions are true).

Proposition 2.1: The altitude of the LZ is within the envelope of the vehicle.

Proposition 2.2: The local wind conditions are within the envelope of the vehicle.

Proposition 2.3: The power margin is within acceptable parameters (nominally +10%).

Proposition 2.4: The decision engine can define a valid ingress route.

Proposition 2.5: The decision engine can define a valid egress/abort route.

Proposition 3.0: The approach and landing can be completed while maintaining suitable conditions (all of the supporting propositions are true).

Proposition 3.1: SWEEP can remain valid during the approach phase of the vehicle (from start to HOGE).

Proposition 3.2: SWEEP can remain valid during from HOGE to landing.

3.3.4 PVS Model

After establishing the top level propositions, we translated them into the theorem proving software package, PVS. PVS is a computer program that contains a theorem prover (symbolic engine that implements the deductive rules of a logic system). It allows the use of precise statements of logic such as lemmas and theorems. Proofs of logic formulas can be mechanically proven using the PVS theorem prover, which guarantees that every proof step is correct and that all possible cases of a proof are covered. Similar to the work performed by Narkawicz and Muñoz [105], all propositions presented were mechanically checked in PVS for logical correctness.

PVS has been used by NASA and other organizations for documentation of requirements for autonomous behavior for FAA certification [72]. The PVS specification (Figure 3.3) is broken down into three sections (similar to the three main propositions). The first deals with the physical size of the LZ (Proposition 1.0). The second deal with the environmental conditions of the LZ (Proposition 2.0). The third with SWEEP remaining valid during approach to landing (Proposition 3.0). Using theorem proving software provides a repeatable, traceable, model of the system's behavior which satisfies the specification. Figure 3.3 is a PVS top level specification that illustrates the requirements for completing the initial SWEEP checks by a decision engine. While this model is not sufficient for formally verifying the specification, we use the model to illustrate how documenting the requirements through a formal process can provide TAEs with artifacts. These artifacts can be additional risk mitigation measures during the certification process for allowing an autonomous system to complete a task currently reserved for qualified pilots.

PVS offers the ability to analyze the propositions listed in Section 3.3.3 within the interactive proving environment. While using the interactive environment, Lemmas can be defined from sections of a PVS specification. An example of this would

```
sweep_to_land: THEORY
BEGIN

suitable:       bool    % LZ is good for size, slope, and is obstruction free
too_small:      bool    % LZ is too small for the vehicle
bad_slope:      bool    % Slope of LZ is unsat for vehicle limits
obstructed:     bool    % LZ has too many obstructions
unsuitable:     bool =  NOT suitable    % LZ is unsuitable

good_conditions:  bool   % Condiitions work for elevaton, wind, Power margin,
                         % ingress route and egress route
too_high:         bool   % Elevation is to to high for vehicle
too_windy:        bool   % Wind exceeds vehicle limits
bad_power_margin: bool   % Less than a 10% power margin
no_ingress:       bool   % no valid ingress route
no_egress:        bool   % no valid egress route
bad_conditions:   bool = NOT good_conditions   % conditions not valid for approach

good_landing:     bool   % successful landing
abt_approach:     bool   % SWEEP goes invalid on approach
abt_landing:      bool   % SWEEP goes invalid on landing
abort:            bool = NOT good_landing   % vehicle aborts on approach

approach: bool = NOT unsuitable AND NOT bad_conditions
valid_landing: bool = approach AND NOT abort

cond_ax1: AXIOM too_small => unsuitable
cond_ax2: AXIOM bad_slope => unsuitable
cond_ax3: AXIOM obstructed => unsuitable
cond_ax5: AXIOM too_high => bad_conditions
cond_ax6: AXIOM too_windy => bad_conditions
cond_ax7: AXIOM bad_power_margin => bad_conditions
cond_ax8: AXIOM no_ingress => bad_conditions
cond_ax9: AXIOM no_egress => bad_conditions
```

Figure 3.3: PVS Specification for SWEEP Checks to Landing Detailing the Decision Process for a Unmanned System to Make a Decision Currently Reserved for a Qualified Pilot

be an evaluation of the environmental condition of the LZ (wind and elevation). If either were outside of the defined parameters of a valid LZ, the selected LZ would be unsuitable due to conditions. An example of this Lemma in PVS can be found in Figure 3.4. For further details on the functionality and utility of PVS, we refer the reader to reference [131].

Theorem provers provide an analytical framework that can completely define the environment the vehicle will be operating in. While the model that is defined is a simplified model of the real world, it is robust enough that flight certification officials

```
%%%% LEMMA 3: Deals with Environmentals %%%%%
Lemma_3: LEMMA
(too_high OR too_windy)
IMPLIES
bad_conditions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Figure 3.4: LEMMA 3 deals with the Environmental Conditions of the LZ: If the Elevation or the Winds are out of Limits the LZ is not Valid Due to Bad Conditions

can use it to justify what the decision engine will not allow the vehicle to do. Thus, allowing the officials to approve the decision engine to exhibit non-deterministic behavior provided the behavior remains within the limits of its clearance envelope.

For theorem provers, assumptions at a top level become requirements at lower levels. The specification outlined in Figure 3.3 has a number of requirements embedded in the assumptions and can be broken up into three categories: LZ Suitability, Environmental Conditions, Status During Movement. Providing all three are satisfied the specification would be valid and verified, and thus provide certification officials evidence of what the system would not do. Therefore it can be used to prove the specified behavior will satisfy the requirements, given the assumptions.

For the PVS model to be a valid artifact for certification officials, it must be representative of actual conditions a vehicle would be faced with. To accomplish this the assumptions built into the top level must be valid. These assumptions are what would define the real world situation. Weather and atmospheric conditions are built into the various states of the model as assumptions. Aircraft procedures and mechanics (such as aircraft size and operational limitations) are also built into the assumptions. Provided the assumptions are valid, a more detailed design implementation is implemented by a more abstract one (the PVS model in Figure

3.3).

Figure 3.5 depicts the results of the PVS model against 11 separate hypothetical LZs. Of the 11 LZs only one is acceptable for landing. LZ 1 is an ideal LZ, as all 10 supporting propositions remain true. LZ 2 through 11 all have one supporting proposition that is false. The PVS specification shows that the final 10 LZs are not acceptable for landing.

	Proposition 1.1 (Size)	Proposition 1.2 (Slope)	Proposition 1.3 (Obstructions)	Proposition 2.1 (Elevation)	Proposition 2.2 (Wind)	Proposition 2.3 (Power Margin)	Proposition 2.4 (Ingress)	Proposition 2 (Egress/Abort)	Proposition 3.1 (SWEEP Approach)	Proposition 3.2 (SWEEP HOGE to Land)	Proposition 1.0 (Suitability for Landing)	Proposition 2.0 (Conditions Suitability)	Proposition 3.0 (Movement)	Successful Landing
LZ 1	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	YES
LZ 2	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	NO
LZ 3	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	NO
LZ 4	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	NO
LZ 5	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	NO
LZ 6	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	NO
LZ 7	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	NO
LZ 8	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	NO
LZ 9	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	NO
LZ 10	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	NO
LZ 11	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	NO

Figure 3.5: Depiction of 11 Hypothetical LZs Against the Propositions Listed in Section 3.3.3 and Later Detailed in the PVS Model

3.4 Protocol

We used the analyzed specification as a baseline for the requirements the decision engine will need to fulfill in executing the CAL/LZ mission. By translating the state machine specification into a flow chart protocol, software designers can develop code based on the analyzed specification. The protocol has been broken into several steps that mirror what a qualified pilot would do while completing the CAL/LZ mission. The protocol translates the propositions into assessments. These steps can be traced directly to the supporting propositions presented in Section 3.3.3:

- **Size Assessment**: Proposition 1.1
- **Slope Assessment**: Proposition 1.2
- **Obstruction Assessment**: Proposition 1.3
- **Wind Assessment**: Proposition 2.2
- **Power Margin Assessment**: Proposition 2.3
- **Elevation Assessment**: Proposition 2.1
- **Ingress Assessment**: Proposition 2.4
- **Escape Route Assessment**: Proposition 2.5
- **Sweep Valid Ingress to HOGE**: Proposition 3.1
- **Sweep Valid HOGE to touchdown**: Proposition 3.2

The protocol depicted in Figure 3.6 satisfies the specification. It serves as an artifact for flight clearance officials when certifying a decision engine to make the decision on where to land a large rotorcraft (a task normally reserved for a fully qualified HAC). The various steps of the protocol can be completed autonomously using current day technology. Size, slope and obstruction assessment can be accomplished via LiDAR and EO/IR vision systems under challenging environmental condition to include degraded visual environments. Wind assessment can be accomplished by comparing the rotorcraft ground track against the current control inputs of the vehicle [132]. Onboard health monitoring systems can be programmed to assess the vehicles performance under all known operating conditions (to include degraded modes possible during a malfunction or emergency situation). The performance characterization can be used during elevation, ingress and escape route assessment.

As stated in earlier sections, this research focused on defining an envelope where the system can exhibit non-deterministic behavior. In the event that the LZ under evaluation does not pass all eight assessments (or SWEEP becomes invalid prior to touchdown) the system would return to the hold/start point and evaluate other possible LZs, in an attempt to find a valid LZ, until it no longer has enough fuel to complete the mission. Provided the LZ in question is within the limits established by the protocol (which defines the envelope where a system can exhibit non-deterministic behavior) it can land autonomously. This can be demonstrated by the system attempting to execute a landing on an empty football field, at sea level, in calm winds conditions. Assuming there were no stands or benches adjacent to the field SWEEP would easily be valid between the 15 yard lines (the goal posts would obstruct from approximately the 15 yard line back to the end of each end zone). When executing the landing, the input conditions cannot guarantee the system would choose one landing spot on the field (as there will be multiple that satisfy the protocol). Under our methodology, the system would be certified to choose its landing point autonomously (cleared to land anywhere on the field that satisfy SWEEP). This would allow the system to exhibit non-deterministic behavior provided SWEEP is valid.

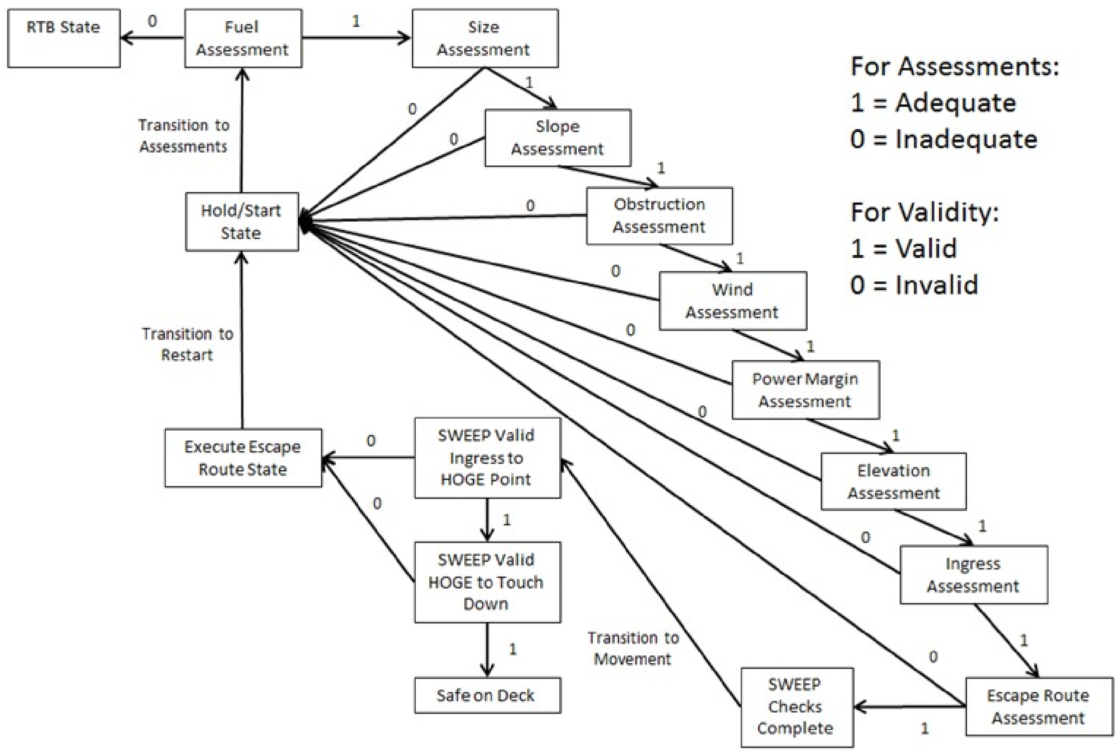

Figure 3.6: Protocol Which Meets the Requirements of the Specification Detailing the Decision Process for a Unmanned System to Make a Decision Currently Reserved for a Qualified Pilot

3.4.1 Evidence Leading to a Naval Flight Clearance

When assessing various LZs the protocol performs eight assessments, each with a binary outcome. These eight binary outcomes translates into 256 possible combinations for each evaluated LZs. A LZ may be large enough for the vehicle in question (so the first value would be a 1), or it may not be large enough (so the first value would be a 0). Of the 256 possibilities, only a LZ that passes all of the assessments (Size, Slope, Obstruction, Wind, Power Margin, Elevation, Ingress, Escape Route) is considered to be a valid LZ which the decision engine can select for landing. The assessments can be linked directly to SWEEP (and the specification) and a limited H-60 clearance envelope:

- **Assessment 1, Size**: Assume H-60, requires a 1.5 rotor arch (75 ft diameter circle)
- **Assessment 2, Slope**: Assume a limited H-60 slope envelope (5 degrees forward/aft, 2 degrees port/starboard)
- **Assessment 3, Obstruction**: Within the circle defined in assessment 1, no obstructions that would hinder a safe landing
- **Assessment 4, Wind**: Assume limited H-60 wind envelope, requires between 2 and 20 kts of head wind and less than 5 kts of crosswind.
- **Assessment 5, Power Margin**: A positive 10% power margin can be maintained through approach to landing.
- **Assessment 6, Elevation**: The LZ elevation is within the operating envelope of the vehicle (below 3,000 ft MSL).
- **Assessment 7, Ingress**: A valid ingress route exists from the start point to the HOGE point.
- **Assessment 8, Escape Route**: A valid escape route exists along the ingress route (to the HOGE point) that returns the vehicle to the reset point.

	Assessment Number							
	1	2	3	4	5	6	7	8
Outcome 1	0	0	0	0	0	0	0	0
Outcome 128	0	1	1	1	1	1	1	1
Outcome 216	1	1	0	1	0	1	1	1
Outcome 240	1	1	1	0	1	1	1	1
Outcome 256	1	1	1	1	1	1	1	1

Table 3.2: Depiction of 5 of the 256 Possible Outcomes of the 8 Protocol Assessments

The results of the eight assessments can be displayed as a binary output. A subset of the 256 possible outcomes of the eight assessments are detailed in Table 3.2. All 256 possible outcomes can be found in Table A.1. If a LZ fails all eight assessments its output would be 00000000 (Outcome 1 in Table 3.2). If it only fails the wind (Assessment 4) its output would be 11101111 (Outcome 240 in Table 3.2). If it only fails the size assessment (Assessment 1) its output would be 01111111 (Outcome 128 in Table 3.2). If it only fails the obstruction and power margin assessments (Assessment 3 and 5) its output would be 11010111 (Outcome 216 in Table 3.2). Only an LZ that passes all eight assessments with an output of 11111111 (Outcome 256 in Table 3.2) would be valid for an attempted landing. After the decision engine chooses a LZ, it would then continuously assess SWEEP until it is safe on deck. While Table 3.2 may seem a trivial contribution, it is in fact considered an artifact that a TAE would use when accepting risk during the flight clearance process [9].

While analytically this appears to be a valid protocol for allowing a decision engine to make the decision currently reserved for HACs consistently, the question remains how can certification officials, within NAVAIR 4.0P, negate the current approved process (CNAF process for naval aviation) where a CO determines they have adequate trust in the HAC prior to full qualification? As a first step, we propose current senior officers become involved early in the process. These officers need to have, or have had, the authority to designate naval aviators as HACs. This is crucial for this effort as it can be used as an additional risk mitigation step to have qualified officers involved in the process.

The protocol (and related artifacts) were also shown to four naval Commanders, all of which have been granted the authority by the CNAF for determining when a naval aviator can be qualified as a HAC. All agreed that assuming the assumptions were valid, the assessments provided would be sufficient to qualify the decision engine to complete the task of landing in a CAL/LZ (a task which currently requires a HAC) safely.

Currently all flight clearances for naval aircraft and subsystems are processed by the airworthiness process using approved V&V techniques/metrics detailed in NAVAIR Manual M-13034.1 [9]. While the evidence presented in this chapter is not currently detailed in that manual, they have been submitted to flight clearance officials for consideration in the next revision of the naval airworthiness process. This may lead to a new process for clearing autonomous behavior under limited circumstances.

3.5 Chapter Summary

To facilitate a flight clearance for a software intensive system, a clear definition of the requirements needs to be agreed upon prior to software development. This chapter presented artifacts for a SOF certification in support of an autonomous controller that is designed to complete the unprepared CAL/LZ mission in a large rotorcraft. The actual path towards this certification does not currently exist.

This chapter was a first step towards a methodology for clearing autonomous behavior to complete the CAL/LZ mission. We defined the requirements normally reserved for a pilot to execute a safe landing on an unprepared CAL/LZ. These requirements were developed through coordination with SOF clearance officials, the naval test and evaluation community, and fleet officials who currently certify pilots as fully qualified. A specification was developed. We then systematically examined the specification in an effort to ensure it satisfies the requirements. Finally we translated the analyzed specification into a protocol and evaluated it against all possible combinations of the conditions of a LZ. The protocol can then be used by software designers when developing the decision engine of the autonomous vehicle. All of the artifacts developed in this chapter can be used as certification evidence for a SOF clearance of autonomous behavior.

Chapter 4: Flight Test of an Autonomous System

Current Safety of Flight (SOF) clearances for unmanned aircraft require a qualified operator who can make decisions and ultimately bear the responsibility for the safe operations of the vehicle. The future of aviation is unmanned, and ultimately autonomous. Yet, a method for certifying an autonomous vehicle to make decisions currently reserved for qualified pilots does not exist. Before we can field autonomous systems, a process needs to be approved to certify them. This chapter analyzes flight test data (both developmental and operational) of an autonomous decision engine selecting an appropriate landing site for a large rotorcraft in an unprepared landing zone. In particular, this chapter focuses on using legacy T&E methods to determine their suitability for obtaining a SOF clearance for a system that possesses autonomous functionality, and focuses on the last three steps of the methodology proposed in Section 1.4. We show that the autonomous system under test was able to complete a mission currently reserved for qualified pilots under controlled conditions. However, when confronted with conditions that were not anticipated (or programmed), the software lacked the judgment a pilot uses to complete a mission under off-nominal conditions.

Many military applications can and have transitioned easily to the civilian sector (e.g. Radio Detection and Ranging (Radar) [78], medevac air ambulance [79], jet engines [80], glow sticks [81], and advanced night vision technology [82]). Therefore, we choose to examine a safety of flight certification for the unprepared (i.e., not an aerodrome or helipad) Confined Area Landing/Landing Zone (CAL/LZ) mission currently carried out the USN and USMC helicopter communities [125]. In an attempt to provide a path forward for certifying autonomy in aviation, this chapter provides insight into the final portion of the certification process: Flight Test (both developmental and operational). We examine flight test data of an autonomous controller as installed on a FAA certified (experimental certification) UH-1 attempting to accomplish the unprepared CAL/LZ mission to determine if the current process can lead to a safety of flight clearance of autonomous behavior. We examined data through the lens of a Developmental Test (DT) program, which is used to determine if the vehicle can satisfy the requirements of the contract for which it was acquired (normally a set of objective measures). Following the DT evaluation, we examined data through the lens of an Operational Test (OT) program, which is used to determine if the vehicle is suitable for the mission for which it was designated under mission representative conditions (normally a subjective opinion of the OT team). Both DT and OT are designed to examine the possible corners of the operational envelope or the edge cases in the software verification [133].

Prior to certification of an autonomous system to complete the CAL/LZ mission, officials need to be provided certification evidence that the system can complete tasks currently reserved for fully qualified Helicopter Aircraft Commanders (HACs) [9]. As a truly autonomous system has never been subjected to formal flight test to support a safety of flight certification, exercising the existing process to evaluate a single mission set will provide significant lessons learned as we transition to more autonomous functionality within aviation. We demonstrate that the autonomous system under test was able to perform the CAL/LZ mission under controlled conditions. However, when confronted with conditions that were not anticipated or programmed (e.g., obstacle types that were not anticipated; compound malfunctions on the vehicle; or changing environmental conditions), its software lacked the judgment a pilot uses to complete a mission under off nominal conditions.

This chapter focuses on flight test of an autonomous system to complete the CAL/LZ mission to determine if it is suitable for a safety of flight certification. This will help build trust in autonomy, as without trust certification officials will be reluctant to grant a safety of flight certification [95]. A simplified version of the steps leading to a safety of flight clearance for an autonomous system to complete the CAL/LZ mission is presented in Figure 4.1. While the flow chart may appear to be a workflow diagram, it is actually a simplified version of the critical path leading to a safety of flight certification. The first step is to determine the requirements the system must complete to accomplish the mission for which it was acquired. Step two involves awarding a contract to a vendor to develop a system that can complete the mission requirements. The vendor will then need to validate the software (ensure the software meets the requirements from the contract), and perform Modeling and Simulation (M&S) as a risk mitigation step prior to flight test. DT will then be performed to ensure the system has completed the requirements of the contract.

Finally OT will be performed to ensure the system can complete the mission under mission representative conditions. Once the system under test has accomplished all the steps, it will be granted a safety of flight clearance. This chapter focus on Steps 5 and 6 of the simplified safety of flight certification process outlined in figure 4.1.

The contributions of this chapter include:

- Development of flight test matrix (one for DT and one for OT) for an autonomous vehicle to complete the CAL/LZ mission.
- Analysis of both DT and OT flight test data of an autonomous vehicle completing a task currently reserved for a qualified pilot (CAL/LZ mission).

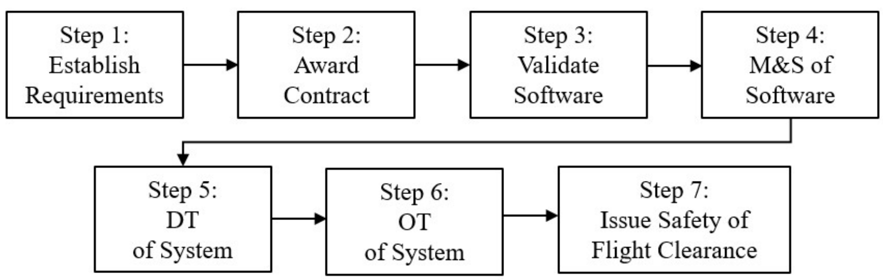

Figure 4.1: Simplified Flowchart Detailing the Steps Leading to a Safety of Flight Clearance for an Autonomous System to Accomplish the CAL/LZ Mission. This Chapter Focuses on Steps 5 and 6

This chapter is structured as follows. Section 4.1 will discuss certifying the CAL/LZ mission, the flight test process, and the system under test (to include a brief overview of the available flight test data). In Section 4.2 DT methods and results are summarized for the system under test. In Section 4.3 OT methods and results are summarized, and a system suitability for the mission is provided. In Section 4.4, we decompose the results of the flight test data for lessons learned regarding flight test of autonomous systems for SOF certification. In Section 4.5, we summarize our finding as they relate to certifying autonomous systems to complete missions currently reserved for qualified pilots.

4.0.1 Current Methods for Flight Certification

Currently, a formalized, or approved, process does not exist for naval aircraft, or aviation systems, that exhibit autonomous behavior (i.e, a system that is able to respond to situations that were not pre-programmed) as there has never been a requirement for one to be developed. Parallel paths are being taken around the world and by other organizations to achieve this goal [134]. However, this chapter focuses on the achievement of a safety of flight clearance for a naval autonomous system. Several possible approaches have been proposed, but none have been vetted through the military, or civilian, flight clearance authorities [43, 92, 110]. The decision space for certifying a vehicle to complete all tasks assigned is extremely complex, which is why this work focused on fight test in support of a SOF clearance of an autonomous controller completing a specific mission: To execute a safe landing of a large rotorcraft (capable of transporting passengers) within an unprepared CAL/LZ. This will enable an exercise of the flight test process for just one mission normally reserved for fully qualified HAC (other missions/tasks would include power line avoidance, see-and-avoid, formation flying, and visual navigation), thus limiting the complexity and scope of flight test.

4.1 Certifying Autonomy for the CAL/LZ Mission

When certification officials grant a safety of flight clearance, they are certifying that if the system were used by a qualified pilot it will be safe and can complete the mission that it was designed for [9]. However, the process of certifying a pilot is a trust process. When certifying a pilot, the commanding officer is putting his or her stamp of approval on a pilot, and they are designating that they trust their judgment when unplanned events occur [10]. By eliminating the pilot from the equation, certification officials need to be able to justify a safety of flight clearance without the benefit of a human in the loop when off nominal condition occur. For the purposes of tractability, we narrow the scope of the problem to a particular flight envelope (i.e., a box) in which the decision engine can exhibit autonomous behavior. This approach will allow certification officials to grant a safety of flight clearance providing the decision engine would not violate one of the limits of the box. We used the Size, Slope, Wing, Elevation, Escape Route, Power (SWEEP) procedure executed by qualified HACs in the USN and USMC [125] (detailed in Section 2.8.2) to define the box for the proposed flight clearance of an autonomous system.

We define a suitable landing as one that satisfies the SWEEP checks performed by qualified HACs. While not all of the steps were specifically programmed into the Tactical Aerial Logistics System (TALOS) (the decision engine that controls AACUS), it is important to understand each component of SWEEP as it relates to the system under test (AACUS/TALOS). In Reference [76] we describe how the SWEEP checklist can be used to define a clearance envelope where a system can be allowed to exhibit autonomous behavior. This can be considered run-time verification, as once the system under test were to reach a edge of the clearance envelop it would revert to known behavior. The components of SWEEP, as it relates to AACUS, are described below:

- **Size**: TALOS used Light Detection and Ranging (LiDAR) to build a 3D image to help determine a landing point free from obstructions and large enough for the vehicle. It was programmed to use a 10 meter diameter as a clear zone for landing. That diameter needed to be an additional 10 meters clear of obstacles (a total of 20 meters from obstructions).

- **Slope**: While TALOS did not specifically determine the slope of a LZ, it used a rough approximation (similar to what a pilot would do) to determine if the slope of the LZ posed an unsafe condition. The slope limits allowed by the controller were more restrictive than the actual limits of the test vehicle.

- **Wind**: TALOS was programmed to continuously evaluate the wind based on the control inputs and the deviations in the ground track (Global Positioning System (GPS) based). This is a standard technique for the test and evaluation of helicopters. On approach it will continue to update its local wind model until it reached 50 ft Above Ground Level (AGL). It then used that wind speed and direction for approach. Prior to landing, the system would maneuver the nose of the aircraft into the wind to minimize crosswind, and maximize headwind.

- **Elevation**: Elevation had a negligible effect on the available flight test data, and was not evaluated. The system under test did not possess a health monitoring system for elevation data. While not evaluated during the test period, the elevation will have a dramatic impact on power available. Providing the data was accurate, it will be a variable for the power portion of the SWEEP checks.

- **Escape Route**: TALOS used the situational awareness obtained by processing the sensor data available to build an escape route. While none of the LZs evaluated required a complicated escape route, one was displayed to the safety pilot and flight test engineer for each approach. During approach, TALOS would monitor the LZ to ensure SWEEP remains valid. If SWEEP became invalid, TALOS will initiate a wave off and fly the escape route back to a hold point.

- **Power**: All of the evaluated test LZs and aircraft configurations accommodated a power margin greater than five percent (a nominal safety buffer the AACUS/TALOS test team put in place). While not evaluated during this test period, it would be a simple limit to place on an autonomous controller.

4.1.1 Flight Test Overview

Flight test is performed on a naval system prior to granting a safety of flight clearance. It is important to understand the purpose of the two types of flight test (DT and OT) as they pertain to granting a flight clearance. The FAA, NASA, and each of the three branches of the United States military have an airworthiness certification process for aircraft. For naval aviation, airworthiness certification authority is delegated to the Naval Air System Command (NAVAIR). When a new capability (i.e., software, weapon or air frame) is acquired, and before naval personnel operate it, NAVAIR must grant a flight clearance (also referred to as a safety of flight certification). Aircraft subsystems, software, components and ultimately the aircraft itself are certified through an established risk mitigation process, the final portion of the process is flight test [9]. Flight test can be further broken down to either DT and OT. The qualification process for naval aviators (pilots) is considered to be a trust process. Unlike the civilian sector, military pilots are trusted by their commanding officers to complete missions critical to national interests. While each pilot is required to log a minimum amount of flight time, and show competency in aircraft procedures prior to qualification, a commanding officer will not designate them as fully qualified until the individual has earned the trust of the commanding officer in their decision making abilities in off nominal conditions [10].

The purpose of DT is to ensure that the system under test can meet the requirements for which it was acquired under (normally a contract). DT is performed by trained test pilots, graduates of an internationally recognized Test Pilot School (TPS). DT points (individual data points required to characterize the system under test during test) are controlled, and designed to determine if the capability meets the individual specifications/requirements from the contract and must be flown by trained test pilots. An example of a developmental test requirement might be "the aircraft will achieve a level accelerated speed of 300 kts at 10,000 ft MSL". This requirement has a clear condition (300 kts at 10,000 ft MSL), and a clear method to achieve the specification (level acceleration). DT also offers an iterative approach to expanding a safety of flight clearance (envelope) by providing data to compare to other types of analysis (such as M&S or wind tunnel data). DT is considered a black or white evaluation of an aircraft against the contract specifications. The test points for DT are typically objective. Once a new capability (i.e., full aircraft, new software, or weapon) has successfully demonstrated that it meets the required DT requirements it can transition to OT.

The purpose of OT is to ensure that the new capability is suitable for the mission it is expected to complete. For a new capability to be deemed suitable (and pass OT) it must be able to perform the mission under mission representative conditions, by fleet representative aircrew. An example of an OT requirement may include "the aircraft must be able to integrate into a multi-plane strike verses a remote target in a contested environment." Modern OT differs from DT in several ways beyond simply the training required for its aircrew. DT is designed to ensure the capability matches the requirements of the contract. OT is designed to ensure that the end user can use the capability to complete its designated mission. It is possible for a capability to successfully pass DT, but fail during OT. This is one of the reasons that United States federal law only requires OT [11]. Unlike the objective evaluation of DT, OT is mainly a subjective evaluation of the system under test's suitability for the mission it is designated for.

4.1.2 System Under Test (AACUS/TALOS) Overview

To evaluate current certification methods for the possible safety of flight certification of autonomy, we required a system that possessed autonomous functionality. In 2017 Aurora Flight Sciences (AFS) developed the TALOS decision engine for the AACUS program under an Office of Naval Research (ONR) contract [132]. AFS installed TALOS on a modified UH-1 which flew under a FAA experimental certificate. The FAA granted the safety of flight clearance for the vehicle with the stipulation that any time the vehicle flew (autonomously or not) a HAC was required to be on board. All flight test data presented in this research was flown by the same experimental test pilot. TALOS used the data available from the onboard sensors combined with the onboard processing power and data buses to build SA of the environment the decision engine would be operating in. The safety pilot (who was a trained experimental test pilot and fully qualified HAC) was required to monitor the systems decisions while the vehicle completed its mission autonomously, and was ultimately responsible for safety of flight.

AACUS/TALOS was designed to execute the Marine resupply mission. We used the available data to analyze the systems performance during the CAL/LZ mission (a submission of the resupply mission). While AFS has published papers within the flight test community, their work focused on how the system was designed, operated and tested [135,136], not on how the flight test results can be used for safety of flight certification of autonomy. Similar work was done by the United States Army in modifying a Black Hawk for field navigation and landing site selection [137].

However the flight test data available from AFS is diverse enough that it can be evaluated under current Department of Defense (DoD) processes [11] for a potential flight clearance of the autonomous controller to complete the CAL/LZ mission. During the test program the safety pilot monitored the system under test while it performed autonomous flight. By utilizing a safety pilot, AFS and ONR were able to examine autonomous functionality despite the lack of certification standards for autonomous vehicles. The 21 flight test events occurred between 11 December 2017 and 23 May 2018. These events were chosen based on the fact that the software controlling TALOS had reached a maturity point where future modifications did not have an effect on how it chose its LZ. The test events also concentrated on the actual landing portion of the demonstration and not the other aspects of the contract. The flights can be broken down to DT and OT like conditions.

The flights supporting the AACUS/TALOS final demo, rehearsals and follow on technology maturation assessment (December 2017 through January 2018) can be seen as DT events. The data set, consisting of six flights concentrated on the system requirements from the contract and the test points were scripted as such. The LZs were located on Quantico Marine Corps Base in Virginia, and were designed to demonstrate the autonomous functionality of AACUS/TALOS. During the DT period, all of the flights were choreographed by the test team to demonstrate the systems ability to satisfy the requirements of the ONR demonstration contract.

The follow on events supporting a large scale field training exercise at Twenty-nine Palms (USMC base in California) can be seen as OT events. During operations in California, 15 flights were flown in the spring of 2018 in preparation for, and in support of, an Integrated Training Exercise (ITX) with actual Marines [135]. The USMC uses Twentynine Palms to simulate real life conditions Marines may find once deployed. The LZs were chosen by actual Marines, to support conditions that can be considered as mission representative. During the OT period, all of the test flights were designed to evaluate the system's capability to complete the assigned task under mission representative conditions.

4.2 Developmental Flight Test of AACUS/TALOS

In this section we further discuss the aspects of DT (Step 5 from Figure 4.1). The evaluation of the objective requirements from the contract are covered in Section 4.2.1. The various test points that will be tracked during the DT period, as well as how the system under test will be characterized, is outlined in Section 4.2.2. A summary of the DT program is provided in Section 4.2.3. Furthermore, in order for a system to pass DT and move onto OT, a positive DT/OT Transition Recommendation (to include the documentation of any deficiencies found during DT) is required. We provide a notional positive recommendation for the system under test in Section 4.2.4.

4.2.1 Requirements of AACUS/TALOS for the Autonomous CAL/LZ Mission

For an autonomous system to obtain a safety of flight certification for the CAL/LZ mission, it will need to demonstrate that it can accurately complete SWEEP checks. As the only parts of SWEEP that were programmed into TALOS were size (to include obstacle detection), slope, wind and escape route, the DT fight test data will evaluate those requirements (elevation and power margin were not evaluated during this test program).

- **LZ Size**: The contract set the requirement for a 10 meter radius (UH-1 rotor arc is 24 ft, 1.6 in), this radius must be an additional 10 meters from any obstacle. The system was required to scan the possible LZ from altitude (approximately 200 ft AGL) and determine if the LZ is large enough for the vehicle. A human pilot uses experience to judge the size of a LZ, but using onboard sensors has the potential of being more exact.

- **LZ Slope**: The contract set the requirement for less than approximately three degrees of slope (actual UH-1 limit is six degrees). The system was required to scan the possible LZ from altitude (approximately 200 ft AGL) and determine if the LZ is within limits. Slope is the most difficult parameter for a pilot to determine from altitude. Often, on approach, a HAC will abort a landing when the slope was not as anticipated from altitude.

- **Obstacle Detection**: The contract requirement was for the system to detect and avoid an obstacle the size of an 18 in pelican case (depicted in Figure 4.2). If a helicopter were to land on an obstacle the risk of dynamic rollover is real. Similar to excessive slope, a dangerous situation can develop if only one skid were to touch down during a normal landing. During the CAL/LZ mission a crew chief actively looks out the side of the helicopter clearing the LZ for the pilots from when the aircraft is over its landing spot through landing. The system was required to scan the possible LZ from altitude (approximately 200 ft AGL) and determine if the LZ is clear of obstacles. The system under test was required to continuously monitor the touch down point for possibly obstructions during approach through touchdown.

- **Wind**: The system under test was able to continuously evaluate the local wind conditions by comparing the ground track of the vehicle against the control inputs. As this is a standard technique for developmental test of helicopters it is not part of this research. As the vehicle begin its approach to landing it stopped evaluating the winds at 50 ft AGL. It then used that wind direction and magnitude to determine if the winds were within limits. Prior to landing, the system would maneuver the nose of the aircraft into the relative wind to limit cross wind and maximize headwind.

- **Escape Route**: The system under test was required to scan the area around the LZ and determine a safe route to a hold point prior to starting its approach for landing. AACUS/TALOS utilized RRT* [121, 122] and the information available through its sensors to build the escape route. If the LZ were to become fouled (something moves into the previously cleared space) or SWEEP were no longer valid during approach (such as an obstacle were to be detected during approach) the system would wave off and fly the escape route to a hold point. In the field, a ground vehicle or wildlife may foul the LZ. Or, once the sensor package was closer to the landing zone it may detect a condition that violates the requirements for a valid LZ.

Figure 4.2: Photo of a Marine Carrying a 24x20x16 in Pelican Case During the AACUS ONR Final Demonstration [138]

4.2.2 Developmental Flight Test Matrix

When preparing for a flight test program, military T&E leadership develop a list of specific test points required to accomplish a test program. Typically these test points are laid out in a easy to follow test matrix. As developmental flight test is resource intensive, leadership will develop test points that are designed to evaluate the edge cases of the system under test. These edge cases typically define the edges of the envelope that will be in a safety of flight certification. These edge cases are typically first identified during risk mitigation M&S prior to flight test (Step 4 from Figure 4.1). The test matrix offers the flight test community a simple to understand status of the test program, and a method to annotate flight test results. To pass DT, the system under test will need to accomplish a minimum of 25 autonomous landings (nominal value we selected for this research), with no safety of flight issues. During the landings, the system must demonstrate that it can select a LZ that is not obstructed and has a slope that meets the requirements of the test program. In addition, the system must demonstrate it can identify an 18 in pelican case in a possible LZ. Finally, during approach to landing, the system must be able to identify an interloper if it were to enter the LZ, abort the landing, and fly to the escape route to the hold point. The flight test matrix, in addition to daily flight reports prepared after each flight, are used by the flight test community to characterize the system under test when they evaluate the systems compliance with the requirements of the contract for which it was acquired. Using the CAL/LZ mission as the foundation for evaluation, the flight test community can help inform certification officials decisions for certifying autonomous behavior. The test matrix for this evaluation can be found in Table 4.1. The columns for Table 4.1 can be described as such:

- **Flight Number - Date**: Specifies the flight test number and date of flight.
- **Size**: Tracks the system's ability to select a LZ that meets the minimum size requirement. During DT this was evaluated by placing obstacles (the test team used pelican cases described in Section 4.2.1) in known locations in the test LZ area to determine if the system can accurately choose a valid landing point (both by the safety pilot in real time and by post flight analysis). Figure 4.3 depicts two LZs. Both photos were taken from the pilot's perspective in

a UH-1, 200 ft AGL over Naval Air Station (NAS) Patuxent River. The left image does not meet the requirements of the contract, the right does.

Figure 4.3: Pilots Perspective of Two LZs Taken From a UH-1 at 200 ft AGL Over the Turf Training Area of NAS Patuxent River [139]

- **Obstruction**: Tracks the systems ability to select a LZ that meets the obstacle clearance threshold (no obstacles larger than an 18 in pelican case). During DT this was by examining the selected LZ to determine that the LZ was not obstructed (both by the safety pilot real time and by post flight analysis). This and the first column of the test matrix will be accomplished by placing test pelican cases around a known location to test the systems ability to choose a valid LZ. Figure 4.4 depicts two LZs that is are obstructed by vehicles.

- **Slope**: Tracks the systems ability to select a LZ that met the maximum slope requirement. During DT this will be evaluated by examining the selected LZ to verify that it meet the slope requirement (both by the safety pilot real time and by post flight analysis). Figure 4.5 depicts a LZ at NAS Patuxent River

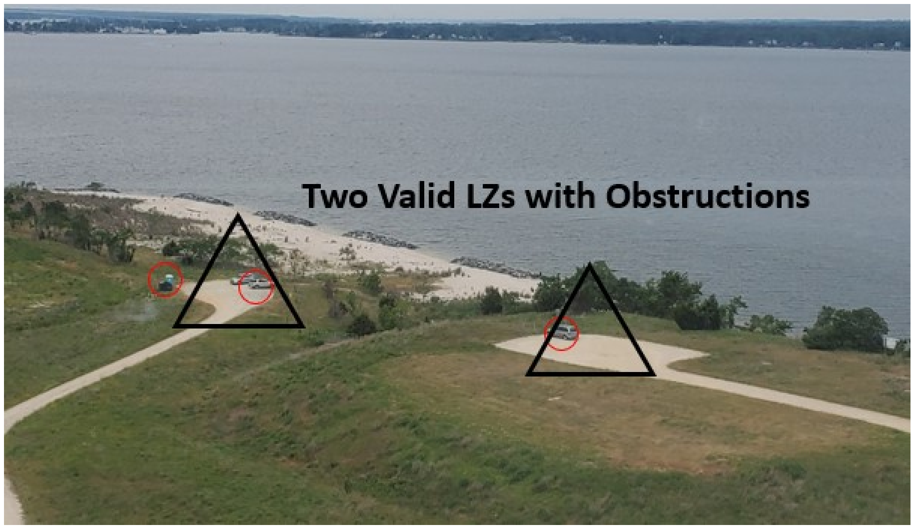

Figure 4.4: Pilots Perspective of Two LZs that Would have Been Valid if the Vehicles Were not Present, Taken from a UH-1 at 200 ft AGL Over the Turf Training Area of NAS Patuxent River [139]

used by the DT community for slope landing evaluation. The photo was taken from the pilot's perspective in a UH-1 200 ft AGL over NAS Patuxent River. The three surveyed LZs have different slopes that test pilots use during flight test.

- **Fouled LZ**: Tracks the system's ability to sense an interloper that fouls the LZ during approach. During DT this will be evaluated by driving a golf cart into the LZ while the system was on approach to landing. Upon sensing the LZ is fouled the system will execute the escape route (which is displayed to the safety pilot prior to approach) and return to the hold point.

- **# Landings and # Aborted**: Tracks safe autonomous landings and aborted approaches by the safety pilot for violation of requirements. To successfully pass DT we stipulated that the system must complete 25 autonomous landings and have zero approaches aborted by the safety pilot for a violation of the

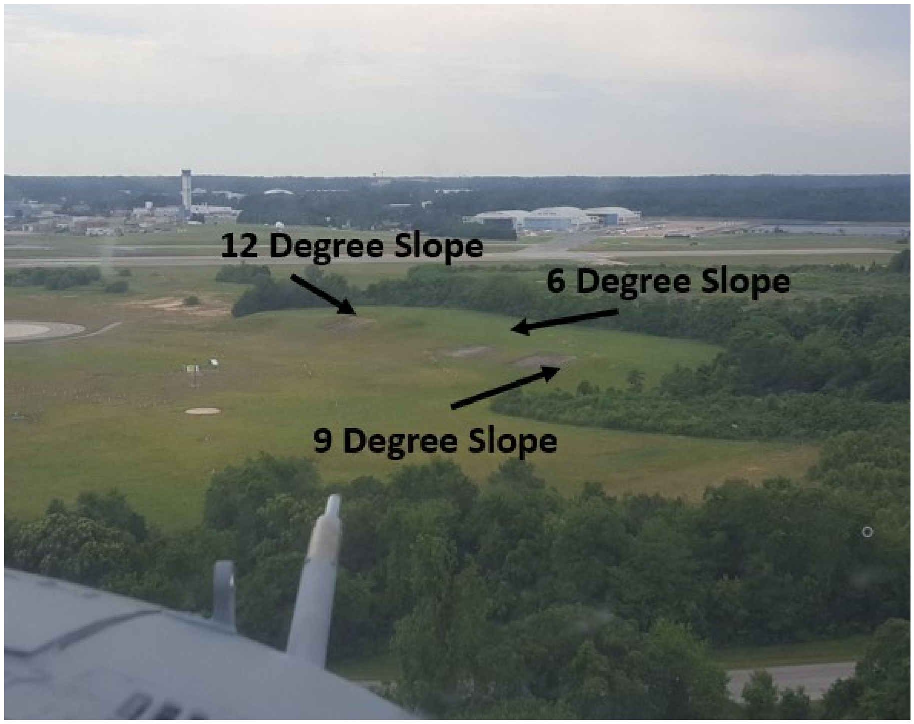

Figure 4.5: Pilots Perspective of Surveyed LZ Used for Slope Landing Evaluation Taken from a UH-1 at 200 ft AGL Over the Turf Training Area of NAS Patuxent River [140]

requirements.

Each DT flight was recorded via the test matrix. The results were evaluated to determine if the system should be recommended for OT, as OT requires substantial investment in resources (both time and money). A system that does not receive a positive recommendation for OT from DT typically does not proceed to the next step until mitigation measures are put in place. Ultimately, the test matrix is used to characterize the system under test.

While the test matrix characterizes the system based on its performance in the execution of planned test points, other items are identified during flight test. Experimental test pilots are trained to find deficiencies in a system. A Part 3 deficiency is considered a nuisance, and is tracked against the system in case there are

Flight # - Date	Size	Obst	Slope	Fouled LZ	# Lds	# Abt
59F096 - 12/11/2017	P	P	P	P	7	0
59F097 - 12/12/2017	P	P	P	P	6	0
59F098 - 12/13/2017	P	P	P	P	5	0
59F100 - 01/22/2018	P	P	P	P	3	0
59F101 - 01/23/2018	P	P	P	N/A	7	0
59F102 - 01/24/2018	P	P	P	P	5	0

Table 4.1: Completed DT Test Matrix of AACUS/TALOS for the Autonomous CAL/LZ Mission (P = Pass, F = Fail, N/A = Not Applicable)

resources (both time and money) available to fix in the future. A Part 2 deficiency is considered an issue with the system that requires human interaction to overcome (such as pressing extra buttons on a flight management system to accomplish the mission). As with a Part 3 deficiency, they are normally tracked for possible correction at a later date. A Part 1 deficiency is one that if not corrected, translates to the system being unable to accomplish the mission, or may result in a mishap. Part 1 deficiencies are typically addressed prior to the system receiving a OT transition recommendation.

4.2.3 Summary of Developmental Flight Test Events

DT of the system under test consisted of six test flights. They were flown as part of the build up to the AACUS/TALOS final demonstration, the demonstration itself, and follow on technology maturation assessment by ONR. All flights took place between 11 December 2017 and 24 January 2018 and were choreographed by the test team to demonstrate the systems mastery of the requirements levied by the contract. Table 4.1 summarizes the six test flights in the test matrix.

During DT, the test conductors used both movable and stationary obstructions to force the system to choose individual LZs that met the requirements of the CAL/LZ mission. When evaluating a LZ, TALOS used LiDAR to build its perception of the LZ. As it approaches a LZ more data becomes available to fine tune its interpretation of the LZ. Figure 4.6 depicts three images showing the perception model of the LZ building as the test asset approaches. The landing area evaluated was a 50 meter radius seven sided polygon. Large obstacles were defined as something with a height of 11 inches. The system would invalidate an area around the obstacle, though not in a circular shape. The shape is elliptical with the long axis parallel to the vehicle's approach path. All images were displayed with north up and the distance to the proposed LZ listed to the lower right of the image. The circle in the center of the image is the desired landing spot from the end user. The colors in the image relate the suitability of the location. Table 4.2 details the color legend for the TALOS produced interpretation of the LZ.

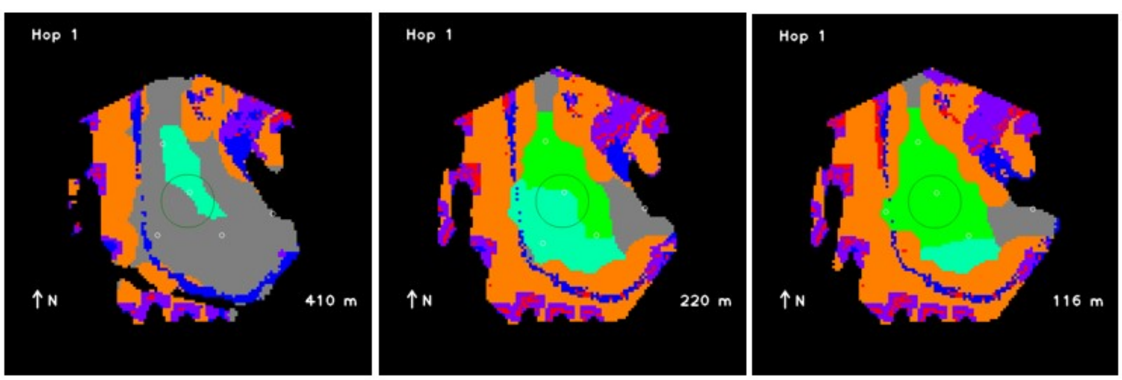

Figure 4.6: TALOS LZ Interpretation from 410, 220 and 116 Meters During Flight 59F097. As the Vehicle Approaches the LZ its Interpretation Become Clearer. [132]

Figure 4.7 depicts the systems interpretation of the LZ for one of the autonomous landings during Flight 59F098 and a image of the test UH-1 immediately post landing. The landing spot was in a field with rolling hills. Figure 4.8 also de-

Color	Meaning
Black	No evaluation performed in the area, or no data available in the area
Gray	No object seen, not enough data to determine if a large size object is present
Yellow	No object seen, not enough data to determine if a medium size object is present
Teal	No object seen, not enough data to determine if a small size object s present
Green	Area is safe for landing, no object seen
Red	Object in this area, not safe for landing
Orange	Too close to an object, not safe for landing
Blue/Purple	Terrain is too sloped or too rough for safe landing

Table 4.2: Legend for Colors in TALOS LZ Interpretation [132]

picts images relating to an autonomous landings during during Flight 59F098, the landing spot was in a simulated Forward Operating Base (FOB), and is considered one of the tougher challenges for the system.

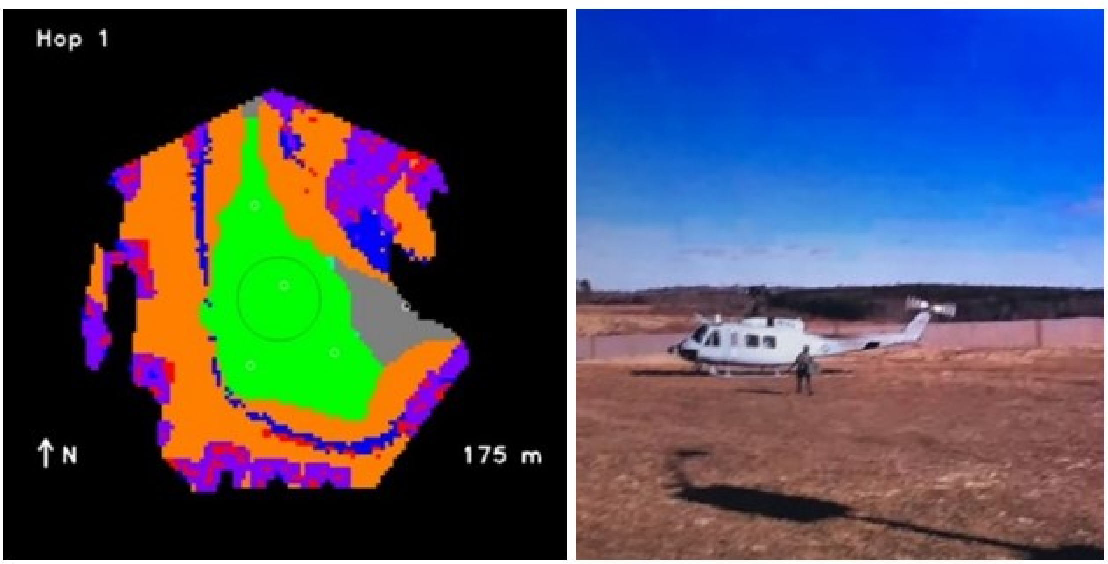

Figure 4.7: Two Images Relating to an Autonomous Landing in a Field During Flight 59F098 Left: TALOS Interpretation of the LZ [132] Right: Picture of the Test Vehicle Shortly After Completing an Autonomous Landing in the LZ Pictured on the Left [141]

To evaluate the system under test's ability to sense an interloper fouling the

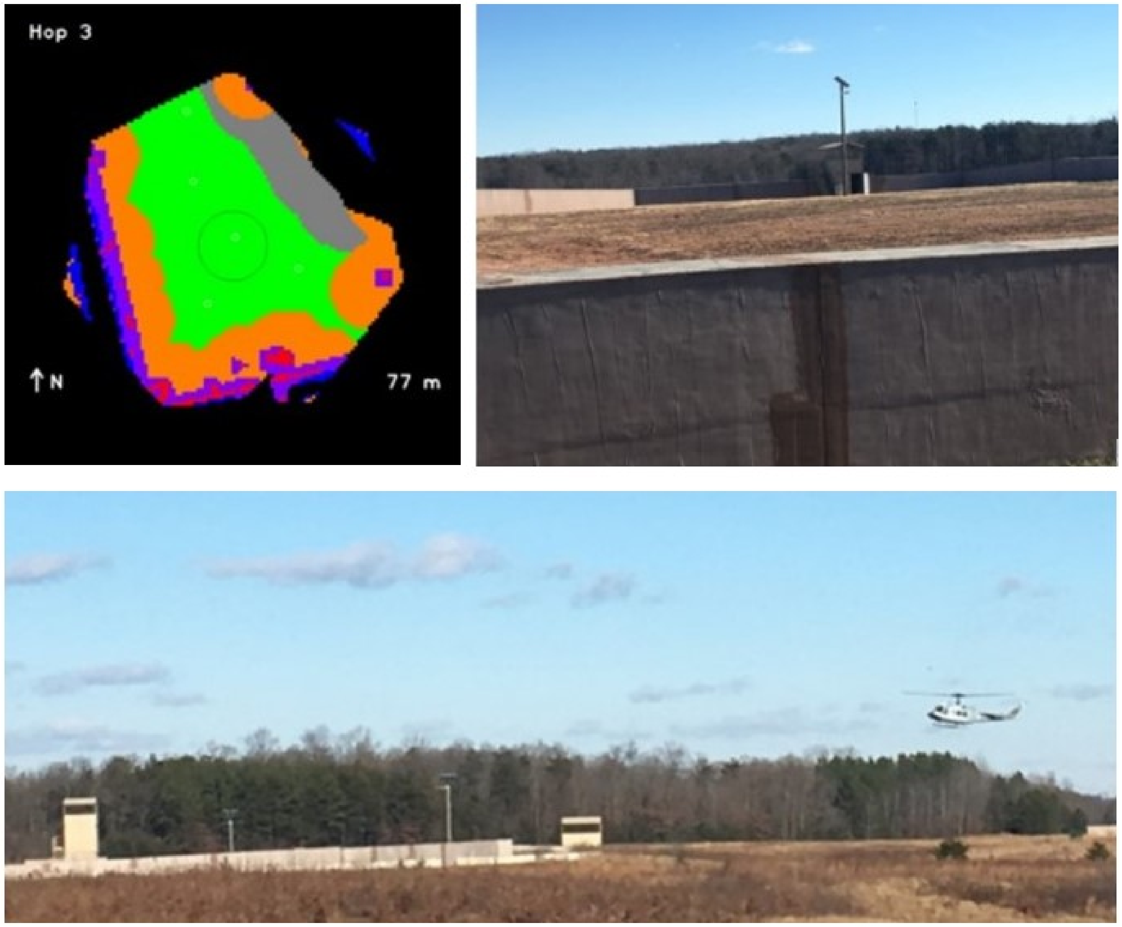

Figure 4.8: Three Images Relating to an Autonomous Landing in a Simulated FOB During Flight 59F098 Top Left: TALOS Interpretation of the LZ [132] Top Right: Picture of the LZ from Ground Level [141] Bottom: AACUS/TALOS Completing an Autonomous Landing in the Simulated FOB [141]

LZ, the test team would wait until the system under test approached the LZ then one of the test team will drive a golf cart into its path. Upon sensing the fouled LZ the system will abort the approach and fly an escape route to the hold point. Figure 4.9 depicts TALOS's interpretation of a LZ before (left image) and after (right image) a golf cart is driven into it. The golf cart is what creates the orange zone at the bottom of the green zone in the second image. This was done to test the wave off functionality of the system.

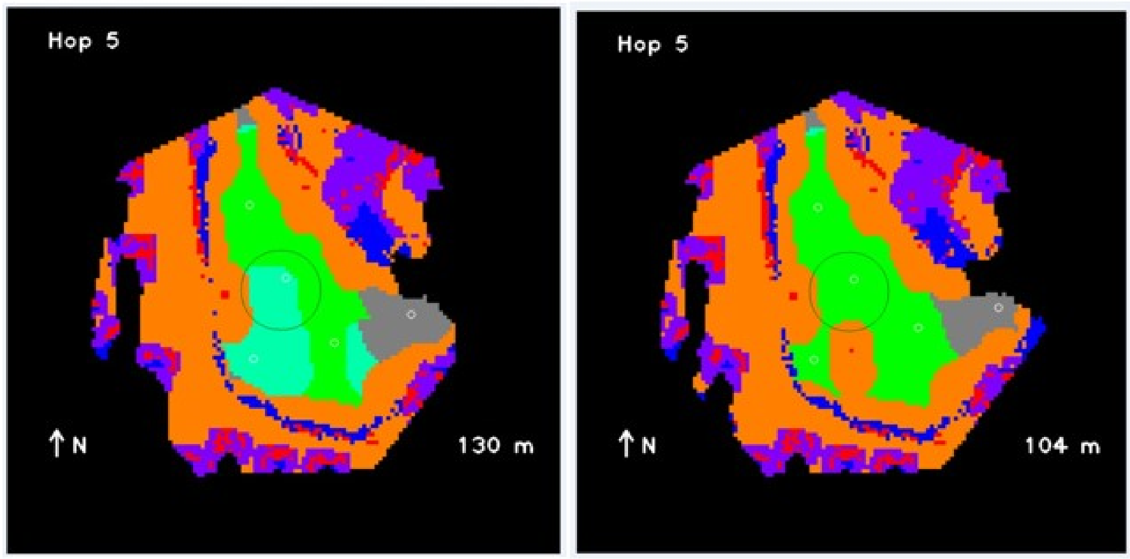

Figure 4.9: TALOS Interpretation of an LZ Before (Left Image) and After (Right Image) a Golf Cart if Driven Into it Testing the Wave Off Functionality on Flight 59F096 [132]

In addition to the test matrix, the safety pilot and test team noted several minor issues during DT. Some of these issues related to the software resiliency, which was not evaluated for the autonomous CAL/LZ mission. Yet, other issues noted by the test team directly relate the system performance. On Flight 59F096 the system selected two landing spots that were not advantageous to the test (one was too close to a road, and one was too close to ground personnel). Although the selected spots met all of the requirements for the system, the safety pilot disengaged the system and selected a more advantageous spot. Also on Flight 59F096 it appeared that the constantly changing cargo load of the vehicle affected the landing performance (both skids did not contact at the same time). On Flight 59F097, while performing an escape route, the vehicle tracked outside of the planned route (yet still safely executed the route) due to the fact that the selected route was not planned to properly match the vehicle's maneuverability. On Flight 59F101 the local wind conditions were more extreme than seen during past test events (winds were 14 gust 19 kts). While the winds were well within the limits of the experiment, the vehicle displayed less than optimal performance (still within prescribed limits).

4.2.4 DT Results and DT/OT Transition Recommendation

Despite the deficiencies noted, the system was able to perform the mission autonomously under the constraints imposed by the test team. We have determined that the system was able to accurately complete the SWEEP checks under controlled conditions and should proceed to OT.

During six DT events the system under test performed 33 autonomous landings with zero safety of flight issues (or violations of the requirements placed by the contract). The system also demonstrated the ability to detect if the landing zone was fouled by an interloper, and execute an escape route to its hold point. However, several deficiencies were identified in the system:

- **First Deficiency**: The system lacks the ability to optimize the landing spot selection (Flight 59F096), once it finds a valid point it for landing it ceased looking for a more advantageous spot (Part 2 deficiency). We recommend that future software loads have a cost function embedded to help solve this problem.

- **Second Deficiency**: The systems actual performance may not be the same as programmed (Part 2 deficiency). We recommend that future software loads have an updated model of the performance of the vehicle.

- **Third Deficiency**: The system lacks a dynamic CG sensing capability which may lead to an unsteady landing (Part 3 deficiency). We recommend that future software loads have an updated CG sensing capability.

- **Fourth Deficiency** During high/gusty wind conditions (yet within the limits of the vehicle/system) the hover and landing performance was safe but not consistent (Part 3 deficiency). We recommend that future software loads have improved gust performance.

4.3 Operational Flight Test of AACUS/TALOS

Unlike DT, OT is not as carefully scripted. During DT the test team was tasked with ensuring the system under test can perform to the requirements that were detailed in the contract. All of the DT LZs were designed to test the capabilities of the system under controlled conditions. Unlike DT, OT flight test is designed to see if the average fleet operator can use the system to perform the mission, and determine if the system under test can perform in a mission representative environment. Operational testers are tasked to determine if the system under test is operationally effective, and suitable for the mission [11]. In Section 4.3 we further discuss the aspects of OT (Step 6 from Figure 4.1). The goals and expectations of the system in OT are covered in Section 4.3.1. The various test point that were tracked during the OT period is outlined in Section 4.3.2. A summary of the OT program is provided in Section 4.3.3. Finally the AACUS/TALOS system suitability assessment (results from OT) are presented in Section 4.3.4.

Late in 2017, AACUS/TALOS showed great promise for autonomy. During several technology demonstration flights the system impressed senior USMC officers. They asked if the system can provide similar results in the field resupplying actual Marine's. ONR and AFS agreed to allow the system to operate at Twentynine Palms, a USMC base in California, during a major USMC ITX. In the spring 2018, AACUS/TALOS flew 15 flights under operationally relevant conditions.

4.3.1 Goals and Expectations of the System in OT

The basic resupply mission is simple: a Marine makes a request for supplies, the request is filled, and a helicopter delivers the supplies to the Marine in the field. AACUS/TALOS was programmed to fly from one location to the Marines location, select a LZ near the Marine, land and allow the Marine to unload the supplies. We evaluated AACUS/TALOS for the final portion of the resupply mission. We evaluated the system under test for its suitability in the autonomous CAL/LZ mission under mission representative conditions at Twentynine Palms Marine Corps Base.

As with DT, we used the SWEEP checklist to determine if the system under test can perform the same actions a qualified HAC would under mission representative conditions. However, during OT we did not evaluate it against black and white requirements. We evaluated it against the safety pilot's (a trained engineering test pilot, and fully qualified HAC) opinions to see if the decisions the system under test made will match that of a fully qualified HAC.

4.3.2 Operational Flight Test Matrix

During the ITX at Twentynine Palms Marine Base AACUS/TALOS was tasked with resupplying actual Marines. As with DT we evaluated AACUS/TALOS for the autonomous CAL/LZ mission (just the landing portion of the resupply mission). However, unlike DT the LZs the Marines chose were not ideal. The obstacles in them were not pelican cases placed by the test team to determine if the system can distinguish a clear LZ that met the requirements of the system. Instead the obstacles were whatever was present in the area where the Marine requested resupply.

For the OT evaluation matrix we once again used the portions of SWEEP that were programmed into the system under test. However instead of evaluating the performance against the requirements of the system (as we did in DT), we evaluated the system against the expert opinion of the safety pilot (a trained engineering test pilot, and fully qualified HAC) while the system performed the autonomous CAL/LZ mission in a mission representative environment. Table 4.3 is a flight test matrix that summarizes operation flight test of AACUS/TALOS for the autonomous CAL/LZ mission and the columns can be summarized as follows:

- **Flight Number - Date**: Specifies the flight test date and flight.
- **Size**: Tracks if the safety pilot agreed with the size of the selected LZ.
- **Slope**: Tracks if the safety pilot agreed with the slope of the LZ.
- **Obstruction**: Tracks if the safety pilot agreed that the LZ was clear of ob-

structions.

- **Spot**: Tracks if the safety pilot agreed with the landing spot chosen by the decision engine.
- **Wave Off**: Tracks if the safety pilot pilot felt the wave off was executed properly.
- **# Landings**: Tracks the number of autonomous landings during the test flight.
- **# Aborted**: Tracks the number of landing aborted by the safety pilot for safety of flight reasons.

In order to successfully pass OT and ultimately be given a safety of flight certification and fielded, the system under test will need to demonstrate under operationally relevant conditions that it can complete the autonomous CAL/LZ mission. Unlike DT where the system merely needed to demonstrate that it met the requirements set in the contract, in OT the system needed to show that it can perform as a fully qualified HAC to be effective and suitable for the mission (a subjective assessment by the OT team).

4.3.3 Summary of Operational Test Events

OT consisted of 15 flights flown between 29 April 2018 and 23 May 2018. They were flown as part of a major field exercise supporting USMC personnel at Twentynine Palms Marine Base. All test flights were flown under mission representative

Flight # - Date	Size	Slope	Obst	Spot	W/O	# Lds	# Abt
59F111 - 04/29/2018	Yes	Yes	Yes	Yes	Yes	2	0
59F112 - 05/01/2018	Yes	Yes	Yes	Yes	N/A	2	0
59F113 - 05/03/2018	Yes	Yes	Yes	Yes	Yes	4	0
59F114 - 05/04/2018	Yes	Yes	Yes	Yes	N/A	7	0
59F115 - 05/06/2018	Yes	Yes	Yes	Yes	N/A	1	0
59F116 - 05/08/2018	Yes	Yes	Yes	Yes	N/A	1	0
59F117 - 05/08/2018	Yes	Yes	Yes	Yes	N/A	2	0
59F118 - 05/12/2018	Yes	Yes	Yes	Yes	N/A	6	0
59F119 - 05/14/2018	Yes	Yes	No	No	N/A	6	0
59F120 - 05/15/2018	Yes	Yes	Yes	Yes	N/A	4	0
59F121 - 05/17/2018	Yes	Yes	Yes	Yes	N/A	4	0
59F122 - 05/18/2018	Yes	Yes	Yes	Yes	N/A	2	0
59F123 - 05/21/2018	Yes	Yes	Yes	Yes	N/A	2	0
59F124 - 05/22/2018	No	Yes	No	N/A	Yes	2	0
59F126 - 05/23/2018	Yes	Yes	Yes	Yes	N/A	2	0

Table 4.3: Completed OT Flight Test Matrix of AACUS/TALOS for the Autonomous CAL/LZ Mission

conditions and not specifically choreographed by the test team to demonstrate the systems mastery of the requirements levied by the contract. The first five flights were system prep flights to understand the new environment. The final 10 were in direct support of the exercise. Table 4.3 summarizes the 15 test flights in the OT matrix.

During the first flight in a mission representative environment, some issues immediately presented themselves. Unlike the LZs of Quantico, those in Twentynine Palms had not been cleared of brush to maximize Marine training. Vegetation in the high desert of California ranges from small shrubs or tumble weed, to shoulder high bushes. The test team used the first flight to judge the effect vegetation has on the system. During 59F111 the system under test had difficulty, in the opinion of the safety pilot, finding a LZ that met its criteria for obstacle clearance. While evaluating four LZs, only two of them met the requirements for the system under test to perform a landing. The safety pilot noted that the UH-1 could have performed a landing, but it would require extensive crew coordination and pilot judgment (these capabilities were not programmed into the system). Figure 4.10 is the TALOS interpretation of one of the LZs and a corresponding google earth image prepared by the test team from Flight 59F111. The safety pilot felt he could land in the LZ, but TALOS couldn't find a valid spot based on the extra safety factor programmed into the system.

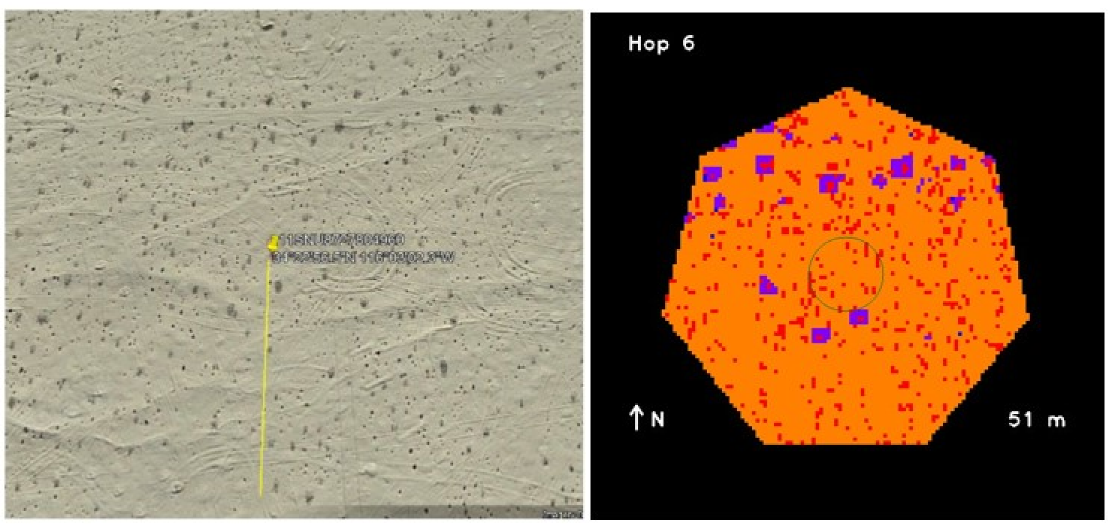

Figure 4.10: Two Images Relating to an LZ During Flight 59F111. Left: Google Earth Image. Right: TALOS Interpretation of the Same Location. TALOS Declared the Location Unsuitable, the Safety Pilot Disagreed [132].

One of the major concerns from AFS and ONR was how the system under test would perform under conditions approaching brown out, where the rotor wash picks up dust when landing in a desert LZ that blocks the aircrew view of the ground when approaching touch down. Several of the LZs chosen during the first five flights at Twentynine Palms were chosen to assess the systems performance in adverse conditions. The fear was that the installed LiDAR could not penetrate dust on landing, and initiate a wave off that is not warranted. No issues were found when operating in near “brown out” conditions. During Flight 59F120 the system was able to complete the CAL/LZ mission despite encountering what the safety pilot considered full brown out (Figure 4.11).

Figure 4.11: System Under Test Performing an Autonomous Landing During Full Brownout Conditions at Twentynine Palms Marine Base During Flight 59F120 [135]

Flight 59F117 was a milestone for the program. It was the first time that the system was used to perform the resupply mission of Marines in the field. The system under test was able to complete the entire mission (to include the CAL/LZ portion of the flight) autonomously.

An issue was found during Fight 59F119. The system under test was to fly to a remote location (that had a dirt runway) for a resupply mission and vehicle refuel. The Marines at the LZ had set up sand bags to indicate to the pilot where to land (a standard operating procedure). However, the system saw the sand bags as an obstruction, and chose a different landing spot. The safety pilot took control of the aircraft and landed on the runway, in the desired location, to facilitate refueling. The other six landings performed during the flight were all accomplished autonomously at other locations with no issues.

Another issue was noted on Flight 59F125. The system under test was directed to resupply Marines in the field with water (mission critical based on the location). Unfortunately, the location the Marines chose for resupply was sub-optimal. The foliage in the area made it difficult for the the system to select a landing point that met the requirements of the programming. The system under test had a requirement for the size of an obstacle. It was programmed to invalidate the area around detected obstacles. A trained HAC would have evaluated the foliage in the area and dismissed some of the foliage as a non factor (yet the system under test identified them as a hazard). Ultimately the safety pilot had to disengage the system and land manually to accomplish the resupply (as the LZ was compatible with the UH-1, just not the requirements programmed into AACUS/TALOS).

4.3.4 AACUS/TALOS System Suitability

The system under test demonstrated that it could complete the autonomous CAL/LZ mission under favorable conditions (i.e., those that were programed into the system). During OT, AACUS/TALOS performed 46 autonomous landings. It also demonstrated extreme promise in controlling a helicopter during brown out conditions. However, under field conditions the experience and training of the safety pilot was required to complete the landing when the obstacles in the LZ were challenging. The system was programmed with a large safety margin, but that margin negated the ability of the system to perform landings in some of the LZs of Twentynine Palms. In addition, some of the LZs chosen by the system under test were not ideal. The vegetation in the proposed LZs had not been completely cleared as it would have been at an aerodrome or helipad, and the safety pilot had to take control and land at a more advantageous spot (mainly when dealing with LZs that required interaction with Marines on the ground). The system also had issues when identifying obstacles that could foul a LZ, as it was programmed to view an 11 in obstacle as fouling a LZ. In the field many of these objects were small shrubs or tumbleweeds. A fully qualified HAC would have identified them as no risk (as the down wash on approach would blow them out of the way). This is also a limitation of the programming in a system that was designed as a technology demonstration, not a system for operational use. While all 15 flights were flown by the same experimental test pilot, the conclusions in this research were formed by a committee of flight test experts who had access to the flight test data.

The results of OT were shared with senior naval officers who currently certify pilots as HACs. They are tasked with certifying the judgment of the pilot to perform critical missions when the conditions were sub-optimal. They unanimously agreed that, as evaluated, the AACUS/TALOS did not meet their threshold as being capable of making decision currently reserved for qualified pilots. When presented with a situation that matches the programming, the system under test was able to complete the mission. However, when presented with a situation that did not fit neatly into the programming the system could not complete the mission.

We found that AACUS/TALOS (as programmed and evaluated) was not effective or suitable for the autonomous CAL/LZ mission. Based on these findings, NAVAIR would not grant a safety of flight certification for the system to perform the mission.

4.4 Analysis of the Test Results as it Relates to Certifying Autonomy

Throughout the 1920s and 1930s, despite meteoric advances in structures, aerodynamics, and propulsion, aircraft handling qualities languished under the conception that it would not be feasible to create objective design standards (satisfying black and white requirements) to achieve a subjective ends (satisfying pilots needs) [142]. The advent of autonomous systems has created a similar daunting task. Currently, certification officials mainly use objective standards to determine if the system can be used by a fully qualified aircrew to complete a mission prior to granting a flight clearance. However, the CO of a squadron uses a subjective measure to determine if a pilot is ready for full qualification. This creates the same problem aircraft designers had for improving handling qualities. The designers of autonomous systems will be given a set of performance specifications which are themselves objective ends. However, the quantities prescribed in specifications, completing a judgment task, requires objective means to an associated subjective end [143]. This research has shown that accomplishing a judgment task (we evaluated the system under test for the CAL/LZ mission) will require new processes, or adjusting current processes to meet the new requirement.

The available flight test data was evaluated under DT like conditions (where applicable) to determine if the contractor was able to build a system to a specification of the contract (show that the decision engine would only land in areas that met the conditions of the contract). It was also evaluated under OT like conditions (where applicable) to determine if the decision engine could execute the task under mission representative conditions. The flight test data was also presented to senior officers who currently certified HACs.

AFS developed the decision engine that enabled the system under test to accomplish the task under controlled conditions. During the notional DT phase of this test program the system under test successfully completed its assigned task 33 times with no issues relating to the landing portion of the test flights. We felt the system under test was able to complete the requirements levied by the contract (objective requirements), and AACUS/TAOLOS would have passed DT and transitioned to OT. However, several of the landings were not optimal. In more than one case, the safety pilot took the controls and delivered the vehicle to a more favorable location. Once TALOS found a location that met the minimum requirements it was programmed to execute, it stopped looking for a better solution. The senior naval officers felt that a HAC needs to use their judgment to pick the best available location for landing. While the system can accomplish the CAL/LZ mission by satisfying the SWEEP checklist and executing an autonomous landing, a more ideal landing point offers an extra buffer of safety. One example was Flight 59F097. During that flight the safety pilot disengaged the system and chose a touch down point to maximize the impending static display following shutdown. The system under test was not aware that a number of high ranking Marine officers were waiting to see the vehicle. Its only concern was finding a valid landing spot. The safety pilot knew that the closer he could land to the distinguished visitors the better. This showed the narrow focus of the decision engine, as changing the programming for touchdown point was not possible between flights. It was not possible to add judgment in the current build of the software.

During follow on testing at Twentynine Palms (considered to be OT data) the system under test was able to complete 46 autonomous landings in mission representative environments. However, the decision engine displayed issues with distinguishing valid landing zones for the test vehicle. This may have been a byproduct of the demonstration program requiring a large safety buffer (much larger clear LZ than required for the platform). The software required a large diameter clear zone for landing. On more that one flight the safety pilot had to take control of the aircraft and execute a safe landing in an area that the decision engine eliminated as a valid LZ. The judgment that senior naval officers rely upon when granting the HAC qualification on aviators is an intangible that is difficult to quantify or program into a decision engine. Ultimately, we determined (with coordination with military certification oficials) the system under test was unsuitable for the CAL/LZ mission and would not be granted a safety of flight clearance as programmed and evaluated.

AFS was able to develop a decision engine and sensor package that could perform the CAL/LZ mission autonomously under controlled conditions. However, when presented with other variables that were not considered, or under field conditions, the decision engine lacked the judgment that a HAC needs to demonstrate to their CO before being fully qualified. This highlights a major issue with certifying autonomous behavior for a safety of flight certification. If requirements are black and white, a simple decision tree can be generated for a decision engine to follow. It is when the decision engine faces off nominal conditions, or unplanned circumstances present themselves, that its actions did not mirror that of a fully qualified HAC.

Academia and industry have proven that they can build aircraft with autonomous functionality. AACUS/TALOS was one such example. However, it was a technology demonstration and was never intended for use beyond that. It was given a specific set of requirements to demonstrate, and it was programmed to do so. This research demonstrated that in order to obtain a safety of flight clearance for autonomous functionality the vehicle must prove that it can perform similar actions to those of a qualified pilot under off nominal, or mission representative conditions.

4.4.1 Insufficient SA that May have Led to a Mishap in an Autonomous Vehicle (Outside of the CAL/LZ Mission)

During the period of performance of the technology demonstration contract AFS demonstrated that the modified UH-1 was capable of accomplishing the assigned mission autonomously. The vehicle was able to use its onboard sensors to build its SA and complete the resupply mission in both controlled and mission representative conditions [144]. However, on at least two occasions the safety pilot had to disengage the autonomous functionality due to SOF concerns when the systems

SA did not match reality.

On 12 December 2017, the AFS UH-1 was on its final preparatory flight (flight 59F097) before the final demonstration flight for senior Marine and ONR officials. The demonstration flight was to be a culmination of the ONR contracted flight test period for the AACUS contract. During one of the flight segments the system was operating autonomously. It lifted off from a simulated FOB in Quantico, VA and proceeded to navigate to its next destination. However, the systems SA did not match reality as it failed to properly evaluate the height of treas within its path. When it tracked away from the FOB, the safety pilot had to disengaged the autonomous functionality as the vehicle was tracking close to some trees (Pilot Quality Rating (PQR) 5 from Figure 5.3). Once past, he reengaged the autonomous functionality [132]. While the inadequate SA only lasted for a few seconds, it could have led to a mishap. Figure 4.12 consists of two images taken at roughly the same point in the scripted demonstration. The left image shows the vehicle approaching the top of the trees just prior to the safety pilot disengaging the autonomous functionality to ensure the vehicle would avoid the trees on 12 December 2017. The right image was taken during the actual demonstration on 13 December 2017 (flight 59F098), and shows the vehicle flying high enough to avoid the trees as the vehicle SA closely matched reality.

Following the final demonstration flight on 13 December 2017, the system was approved for follow on T&E in a mission representative environment. Prior to the follow on T&E, ONR and AFS performed a number of technology maturation flights. On 23 January 2018 (flight 59F101) the system again demonstrated an issue with

Figure 4.12: Two Images from the System Under Test Taken at Roughly the Same Point During the Scripted Demonstration. The Left Image was Taken on 12 December 2017 and Depicts the System Approaching a Position Where the Safety Pilot Felt May have Been Unsafe (Close to the Trees). The Right Image was Taken on 13 December 2017 at Roughly the Same Place. However, in this Case the SA of the System Under Test Matched Reality as it had Climbed to a Safe Height to Avoid the Trees [132].

SA. During its initial takeoff the planned route would have transited through some trees (PQR 6 from Figure 5.3), depicted in Figure 4.13. The safety pilot disengaged the autonomous functionality, flew past the trees, and reengaged the autonomous functionality [132]. Shortly thereafter the system operated autonomously for over half an hour completing the resupply mission under controlled conditions [132]. Again, the inadequate vehicle SA only lasted for a few seconds, but it would have led to a mishap if the safety pilot had not intervened.

Post flight analysis of the 12 December 2017 and 23 January 2018 flights discovered a issue within the system architecture. The issue was discovered empirically during testing. While laser returns came back very quickly and identified a physical presence to the raw sensor processing software, that information then had to populate the height map internal to TALOS. Once added to the height map the

Figure 4.13: Image From the System Under Test During the 23 January 2018 Flight. At the Depicted Point During the Flight the Systems SA of the Environment Didn't Match Reality. The Autonomous Functionality would have Flown Through Some Trees and Caused a Mishap if the Safety Pilot Didn't Take Control of the Vehicle [132].

trajectory planner needed to build a route to bypass the obstacle. This issue was particularly evident if the aircraft made a turn towards a departure heading once in a hover, because the obstacle would be initially outside of the LiDAR's field of view. The time to sense an obstacle, populate the height map, and plan a clear route can be measured in seconds (a fully qualified helicopter pilot could complete the task in a fraction of that time). As the system was preparing to operate in mission representative environments, the decision was made to simply add an extra time delay to allow processing to happen on takeoff [132]. The test team felt the extra delay would help during the upcoming T&E where they anticipated degraded visual environments under mission representative conditions [144]. AFS determined that the deficiency was most likely due to the system being designed for experimentation at the unit/module level rather than for overall system level performance [132].

Once the delay was added to the system, the issue was not seen again. However the fact that by simply adding a delay (giving the system more time to process it sensor data) to the system seemed to correct the inadequate vehicle SA implies that there may be a relationship between sensor degradation and SA in an autonomous system.

4.5 Chapter Summary

The existing paradigm for test and evaluation is to define what a system will do given a set of input parameters. Prior to a safety of flight clearance, certification officials currently need to understand how a system will react when used by a fully qualified pilot/operator when completing a mission (such as the CAL/LZ mission). By removing the pilot/operator (for autonomous systems) we believe that we can obtain a flight clearance for autonomy based on what the system will not do. To define a box where a system can be allowed to exhibit autonomous behavior, we used the SWEEP checks performed by the USN and USMC helicopter communities. We were able to evaluate flight test data of an autonomous system (the AFS AACUS/TALOS UH-1) completing the CAL/LZ mission under controlled conditions (DT) and under mission representative conditions (OT).

Between the AACUS/TALOS final demo (to include the rehearsals) and the ONR Technology Maturation assessment the decision engine under test demonstrated 33 autonomous landings, and several wave off approaches based on a fouled

LZ. These flights could be considered DT events as the conditions were controlled to demonstrate the objective requirements of the contract for which the system was acquired under. During these test flights the decision engine was able to define a safe landing spot that met the constraints of the contract. Therefore, the decision engine would have met the objective requirements of DT. However, several deficiencies were noted with the system. The most troubling was that once the system picked a landing point that satisfied its programming, it did not continue looking for a more advantageous spot. Yet, based on the performance of the system under controlled conditions, it would have passed DT and been recommended for OT (to be evaluated under mission representative conditions).

During the ITX evaluation period, the AACUS/TALOS system was used in a mission representative environment (OT), Twentynine Palms Marine Base. During 15 flights, the system under test executed 46 autonomous landings in environments similar to those that would be needed to execute the CAL/LZ mission to resupply Marines in the field. However, the OT evaluation is a subjective test. The purpose of which is to determine, to the subjective opinion of the OT organization, if a standard fleet user can use the system under test to complete the desired mission under mission representative conditions. While the vehicle demonstrated the ability to stay within the clearly defined envelope, several decisions made by the vehicle were in contrast to what a qualified HAC would have made. None of the decisions would have resulted in an unsafe condition. However, the results of an OT report on the data available would have found the system under test unsuitable for the autonomous CAL/LZ mission as programmed.

This chapter used legacy test procedures for the evaluation of the system under test. While the procedures provided data on the system and may be a valid method to test an autonomous system, they did not provide a method to correct issues early in the development cycle. Once a system reaches flight test it is extremely difficult to fix the system and still meet deadlines. If an autonomous system were to be certified safe for flight, the most important step will be to ensure the requirements are specified in such a manner that system developers can program in the ability to cope with off nominal conditions.

Academia and industry have demonstrated that they can develop a system that can exhibit autonomous behavior while completing a mission normally reserved for qualified pilots under controlled conditions, AACUS/TALOS is one such system. However, when confronted with conditions that were not programmed into the decision engine the actions of the autonomous system did not match that of a fully qualified pilot. By using the SWEEP checklist as a guarantee of what the system will not do, flight clearance officials can grant a safety of flight clearance for autonomy. However, prior to authorizing that clearance the software package to complete tasks that require a pilot's judgment the system needs to demonstrate it can accomplish the mission under controlled and off nominal conditions.

The existing data on the AACUS/TALOS system is promising for the future of unmanned vehicles supporting the CAL/LZ mission. However, the narrowly defined focus of the current AACUS/TALOS architecture is inadequate for the mission need.

Chapter 5: Developing a Objective Measure for a Subjective End

Pilots use Situational Awareness (SA) to make appropriate aeronautical decisions. Autonomous vehicles will not have a human pilot, or operator, in the loop when off nominal conditions present themselves. They will rely on sensors to build SA on their environment to make sound aeronautical decisions. As their sensors degrade, we hypothesize a point exists where the SA those decisions are based off will be inadequate for sound aeronautical decisions. We show that this point can be identified through Modeling and Simulation (M&S) of a simple sensor network to complete a task currently reserved for qualified pilots. This chapter highlights the process of determining an objective measure for this subjective end, and relates it to a possible Safety of Flight (SOF) certification for an autonomous system to perform tasks currently reserved for qualified pilots. This chapter and focuses on the fifth step of the methodology proposed in Section 1.4.

Pilots are trained to use their senses and experience to build their SA while flying to enable them to safely accomplish their mission and make sound aeronautical decisions. The future of aviation is unmanned and ultimately autonomous. However, by eliminating the human pilot (or operator in the case of unmanned aircraft) we will be eliminating the SA that is currently required to safely accomplish a mission when off nominal condition present themselves. This chapter defines an objective relationship between an autonomous vehicle SA while its sensors degrade and the ability to accomplish a task currently reserved for qualified pilots.

The first step in evaluating if the choices an autonomous system makes match that of a qualified pilot is to determine if the SA of the vehicle matches reality for the environment it is operating in. As both military and civilian pilots use SA, we elected to study SA of a autonomous vehicle completing a task currently reserved for qualified military pilots. In prior chapters we examined obtaining a SOF certification for a system that displays autonomous behavior. The underlying focus of our research relates to certifying an autonomous naval system to perform tasks currently reserved for qualified pilots. However, during our research we determined that in order for autonomous behavior to be certified it would need to demonstrate that it can make decisions similar to fully qualified pilots [76, 144], to include situations where a fully qualified pilot makes decisions based on their SA when encountering off nominal or unexpected conditions (such as degradation in the quality of information available to them).

Typically aircraft are designed to objective measures (i.e., maintain a desired speed at a desired altitude). During the certification process the system under test will be required to demonstrate it can complete a subjective end (i.e., integrate with currently fielded systems). It is extremely difficult for designers to build an aeronautical system to accomplish a subjective end without an objective measure. This research focuses on developing a relationship between sensor performance degradation and vehicle SA in an attempt to establish an objective measure that can be provided to designers and certification officials for autonomous air vehicles to complete a task currently reserved for qualified pilots. This will enable certification officials to trust that an autonomous system has a clear understanding of the environment it is currently operating in, and will make appropriate aeronautical decisions (based off its programming) similar to those of a fully qualified pilot. We develop an objective relationship between sensor degradation/error and SA withing a M&S environment. First, we develop a scenario where an autonomous vehicle is reliant on its sensors to build its SA. The scenario was build in such a way that the only factors effecting the SA of the vehicle were the accuracy of its sensors. We then degraded those sensors to a point where the decisions it makes are no longer sound aeronautical decisions. And as a result of this work, two inequalities (objective measure) are defined for when an autonomous vehicle has sufficient SA (subjective end) to make decisions currently reserved for qualified pilots.

This chapter is structured as follows. In Section 5.1 we provide an overview of SA and discuss the issue of defining an objective measure for a subjective end to aircraft designers. We discuss the evolution of Handling Qualities (HQ) specifications to include the use of the Cooper-Harper Rating (CHR) scale which enables an objective value for a subjective task. We also demonstrate how the scale was modified for the evaluation of a highly automated task and later used during the Test and Evaluation (T&E) of an autonomous aeronautical system. In Section 5.2 we begin our use of DOE and a M&S environment to develop a quantitative relationship between sensor degradation and autonomous vehicle SA. In Section 5.3 we develop an objective measure for an autonomous vehicle's SA to accomplish a task, currently reserved for a qualified pilot, as it's sensors degrade. In Section 5.4, we summarize our findings as they relate to certifying autonomous systems to complete tasks currently reserved for qualified pilots.

The contributions of this chapter include:

- The development of an objective measure for autonomous vehicle SA that accounts for sensor degradation.
- The development of a scenario, within a Department of Defense (DoD) recognized M&S environment, that specifically evaluates the effects of sensor degradation on error distance of a fused track of a threat aircraft.
- The use of Design of Experiments (DOE) to determine the effects of sensor degradation and produce predictive equations for the error distance of the fused track.
- Use of Subject Matter Expert (SME) opinion to define the point at which (within this scenario) the fused error distance is inadequate to make a decision currently reserved for qualified pilots.

5.1 Overview of SA and Developing a Objective Measure for a Subjective End

This chapter focuses on developing a relationship between sensor performance degradation and vehicle SA (considered largely a subjective opinion). This is in an attempt to establish an objective measure that can be provided to designers, and certification officials, of an autonomous vehicle to complete a task currently reserved for qualified pilots. Some related work is mentioned in Section 5.1.1. Translating a subjective end into objective measures is not a new concept. Section 5.1.2 details how test pilots translate their opinion of the flying qualities of an aircraft into measures engineers can use to help improve the performance of the control laws via established rating scales. Section 5.1.3 details how the rating scales outlined in Section 5.1.2 have been adapted to allow a test pilot to describe the behavior of an aircraft during a highly automated task (landing a high performance jet aboard an aircraft carrier "hands free"), and later for the evaluation of an autonomy demonstration vehicle. Similar to the use of ratings scales detailed in Section 5.1, designers and certification officials will find an objective measures for subjective ends invaluable for evaluating the SA on an autonomous system.

5.1.1 Current Methods for Flight Certification and SA

This chapter focuses on the SA of an autonomous system as its sensors degrade. This will help build trust in autonomy, as without trust certification officials will be reluctant to grant a SOF certification for a system to operate without a pilot, or controller, in the loop [95]. Currently a formalized/approved process does not exist for naval aircraft/systems that exhibit autonomous behavior (the system is able to respond to situations that were not explicitly pre-programmed) as there has never been a requirement for one to be developed. Several possible approaches have been proposed for autonomous control (dealing with type of controller applied to the vehicle [110], updated the path planning based on sensor input [92], dynamically re-plan the flight path via adaptive controllers [43]) but none dealt with sub-optimal sensor performance or were vetted through naval flight clearance authorities.

An understanding of SA as it relates to aviation is critical to understanding how it will relate to the certification of autonomy. One of the most commonly accepted definitions for SA is "the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future [77]". During flight school, student naval aviators (pilots) are taught that SA in aviation is being able to accurately diagnose what is happening around them and predict what will happen in the immediate future, thus enabling them to perform the assigned mission safely. Students with high SA are able to "stay ahead of the aircraft", while students with low SA tend to seem to be "holding onto the stab" during flight. From their first flight, aviators learn to use every available resource to develop their SA (e.g., radio calls, aircraft instruments, visually scanning outside of the aircraft, onboard radar, Electro-Optical/InfraRed (EO/IR) sensors and seat of the pants feelings). Prior to obtaining full qualification, a naval aviator will have proven to their Commanding Officer (CO) that they can develop their SA to an appropriate level that they can safely complete their assigned mission during off nominal conditions [10]. The measurement of SA has proven to be an intangible, and largely subjective. Pilots quickly learn that the only way to know exactly the level of their current SA is when they realize that they have none. When a pilot's SA is high (i.e., they have an accurate understanding of the environment they are operating in) they can make sound aeronautical decisions. However, when a pilot's SA is low (which they may or may not know at the time) their aeronautical decisions may not be sound.

Autonomous vehicles will use their sensors to build SA of their environment. When sensors are operating at 100% the SA they provide the vehicle should be adequate to make sound aeronautical decisions. However, at some point of sensor degradation the SA provided will no longer match reality. The advent of unmanned aerial vehicles (UAVs) has sparked a increase in research within the academic and flight test communities. When programming UAVs with automation (such as what actions to take in the case of lost link), or autonomous functionality (allowing the vehicle to make decisions based off the conditions they sense), it is vital for the system to be able to safety complete the assigned mission. Sensors are typically installed to inform the operator, or system, of the conditions the vehicle is operating in. These systems could be as simple as a camera, or as complicated as a fusion of multiple sensors. Increases in processing power has enabled these vehicles to perform simple missions (e.g. collision avoidance and visual navigation), under fairly static conditions, providing they have access to sensor inputs. However, when a human pilot realizes that there may be an issue with their SA they have the training and experience to rely on various inputs to diagnose their current interpretation of reality. Unless a system is programmed to react to sensor degradation, certification officials will hesitate to allow the system to make decisions based on the sensor input without a human in the loop to ultimately shoulder the responsibility for the air vehicle.

For a pilot to make sound decisions, they need to have a clear understanding of the situation/environment they are operating in [145]. Teaching a prospective pilot how to develop their SA and knowing when to question their perception are critical portions of flight training [10,146]. Researchers have spent decades developing models and methods for evaluating a pilots SA (highly subjective) during flight and translating it into an objective measure [145–149]. Two methods that have provided ample data for research involve freezing a simulation and asking questions relating to the pilots SA or asking questions of a pilot post mission [146,148]. Yet, neither of these methods allow a pilot to rate their SA in real time to determine when it is lacking. One school of thought was to offer pilots more information to help build their mental picture. Modern aircraft can present a massive volume of data to the pilot. However, this overload of information has a tendency to detract from the pilot's SA and work has been done to optimize how the information is presented [150,151].

As UAVs have become commonplace in aviation, the issue of sufficient operator SA has become a hot button issue. How can an operator maintain appropriate SA to their air vehicle when they are not actually in the vehicle (as a pilot is for manned aviation)? Several papers have been published regarding increasing the SA of a detached operator as to the environment the vehicle/system is currently operating in (to include the status of the vehicles subsystems) on earth [152–159] and space [160,161]. As vehicle based computing power has increased research has been accomplished to demonstrate that a vehicle can navigate via onboard sensors (without direct operator direction) [162–167]. It has been proposed that as the level of autonomy increases, the required level of SA for the human operator will decrease and the required SA of the air vehicle will increase [168,169]. However, the current body of work lacks the ability to demonstrate to SOF clearance officials the ability of an autonomous system to maintain SA while completing its assigned mission as sensor performance degrades.

5.1.2 Development of a Objective Measure (Cooper-Harper Scale) for a Subjective End (Handling Qualities)

This subsection is used to illustrate how an objective measure (the dynamics of a aircraft, e.g. short period) can be used to accomplish a subjective end (CHR of the aircraft handling qualities). Throughout the 1920s and 1930s, despite meteoric advances in structures, aerodynamics, and propulsion, aircraft HQ languished under the conception that it would not be feasible to create objective design standards (satisfying black and white requirements) to achieve a subjective ends (satisfying pilots needs) [142]. Aircraft designers did not have a clear direction for what equated to positive HQ. By the 1940s the first HQ specifications were established, enabling aircraft designers to build aircraft that would have satisfactory HQ for pilots. The specification dealt with both longitudinal and lateral characteristics for the full range of aircraft configurations. One example of an objective measure that led to favorable HQ (subjective end) was placing a quantitative upper limit on the absolute value of the stick-force gradient [142]. For further details on the establishment of objective measures for subjective ends for the first HQ specifications we refer the reader to Chapter 3 of reference [142].

Determining an aircraft's HQ is a daunting task, as different pilots may have different opinions on this subjective judgment. During Test Pilot School (TPS), future test pilots are trained on classical test techniques to evaluate aircraft. One of the corner stones of this training is the Cooper-Harper Handling Qualities Rating Scale (Figure 5.1) as it forces a pilot to make a series of relatively unambiguous decisions to arrive at a rating of the current HQ of the aircraft [170]. CHR is the basis of the US flying qualities Military Specification (Mil-F-8785B, later superseded by 8785C [171]), and divides the pilots opinion of the aircraft HQ into four levels. Level 1 is satisfactory. Level 2 is not satisfactory HQ, but performance is satisfactory. Level 3 includes maximum workload to get adequate performance (and deals with aircraft controllability). Level 4 is uncontrollable [170–172]. CHR 1-3 equate to Level 1 HQ. CHR 4-6 equate to Level 2 HQ. CHR 7-9 equate to Level 3 HQ. CHR 10 equates to Level 4 HQ. Figure 5.2 is from Mil-F-8785C and illustrates how an objective measure (aircraft characteristics, short period dynamics) can be related to a subjective measure (flying quality level). For further details on aircraft HQ we refer the reader to Reference [172].

5.1.3 Cooper-Harper Adjusted for Confidence in Automation

CHR allows the flight test community a method of achieving repeatable results for HQ evaluations. The scale was later used as the blueprint for a rating scale that measures a test pilots confidence of a vehicle accomplishing a highly automated task, landing high performance jet aircraft on the pitching deck of an aircraft carrier without pilot input [2]. The Precision Approach and Landing System (PALS) installed

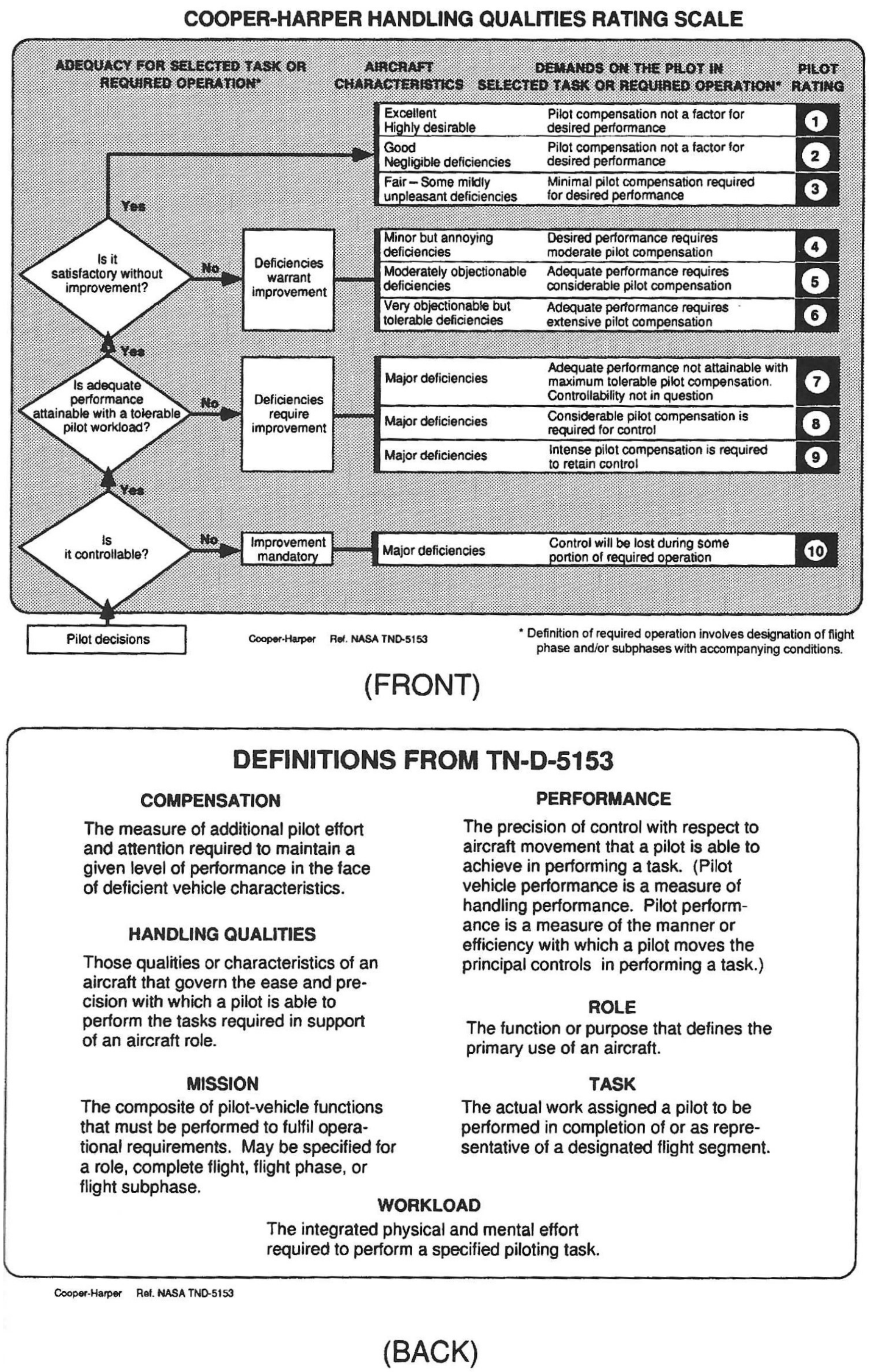

Figure 5.1: Cooper-Harper Rating Scale (Card Used by Handling Qualities Engineers and Test Pilots) [170, 172]

on United States Navy (USN) aircraft carriers allow a pilot to “couple” with the ship and land during adverse conditions (e.g., extreme weather, or when the pilot is

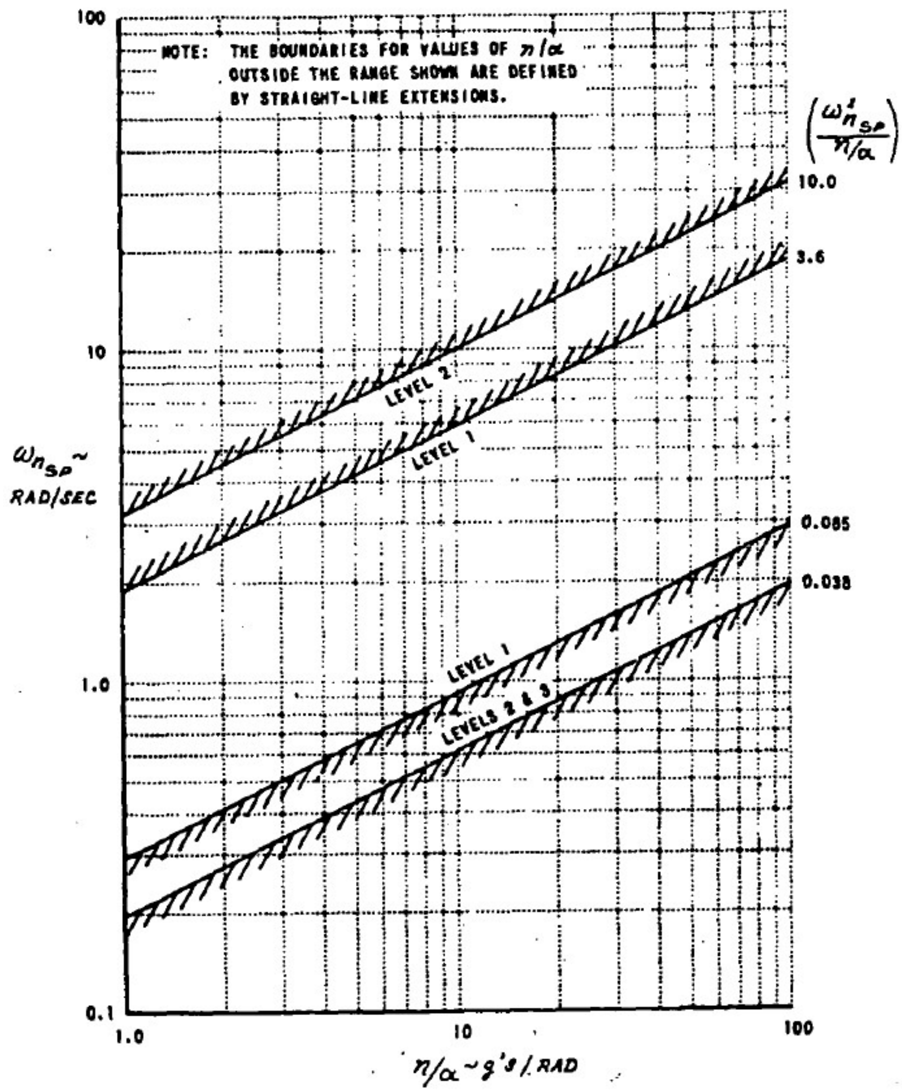

Figure 5.2: Relating Short Period Aircraft Dynamics to Aircraft Handling Qualities Levels During Nonterminal Flight Phases that are Normally Accomplished Using Gradual Maneuvers and Without Precision Tracking, From Mil-F-8785C [171]

unable to perform an arrested landing on their own). Figure 5.3 is the PALS/Pilot Quality Rating (PQR) used during PALS certification testing. PQR allows a test pilot to put their subjective opinion (confidence in the system at accomplishing a task) into a objective measure (PQR rating). For certification, a PALS system must return a PQR of 3 or less. The PQR scale gives PALS engineers an objective measure (PQR rating) for a subjective end (pilot confidence in the system) to use as they adjust the parameters within the system during certification testing [86].

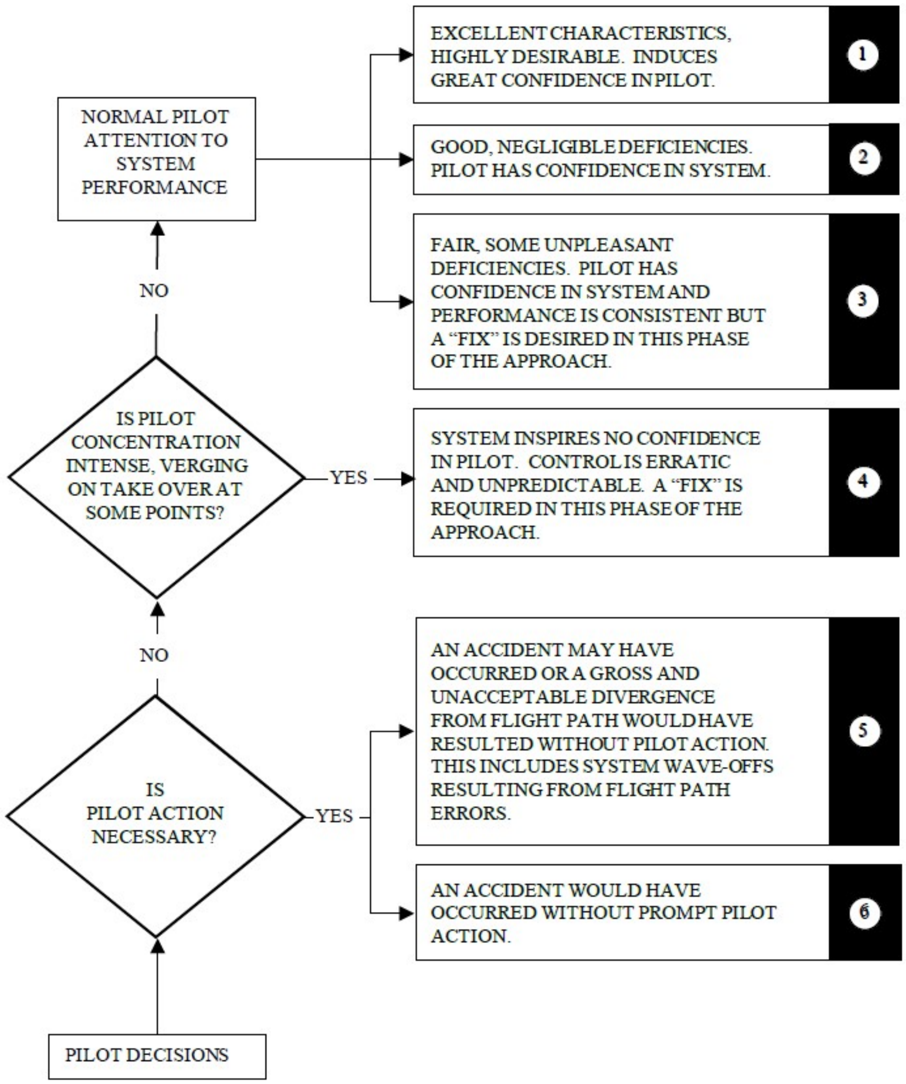

Figure 5.3: PALS/Pilot Quality Rating Scale, Allows Test Pilots to Objectively Gage their Confidence in the System Under Test. Developed for PALS Testing [2,86], and it has Been used for Evaluation of Autonomous Systems [136]

PQR was later adopted by a flight test team evaluating an autonomous controller completing the United States Marine Corps (USMC) resupply mission in a optional piloted UH-1 helicopter during an autonomy demonstration program. The vehicle was able to use its onboard sensors (Global Positioning System (GPS), Light Detection and Ranging (LiDAR), EO/IR cameras) to build its SA and complete the resupply mission in both controlled and mission representative conditions [144]. However, on at least two occasions the safety pilot had to disengage the autonomous functionality due to SOF concerns when the systems SA did not match reality (detailed discussion can be found in Section 4.4.1). The first instance required the safety pilot to take control when the vehicle was tracking dangerously close to trees (PQR-5). The second instance was on a later test flight and required the safety pilot to take control to avoid flying into trees (PQR-6). As these events occurred late in demonstration program, the test team decided to add a delay to the system before it started moving (to allow the onboard processors to spend extra time building is SA of the environment). Once the delay was added to the system, further issues with path planning were not seen [132]. However the fact that by simply adding a delay (giving the system more time to process it sensor data) to the system seemed to correct the inadequate vehicle SA implies that there may be a relationship between sensor degradation and SA in an autonomous system.

5.2 Problem Formation

In Section 5.1.3 we identified a possible relationship between sensor performance and vehicle SA in an autonomous system. In Section 5.2 we develop a relationship between sensor degradation and vehicle SA in a M&S environment through the use of DOE [173]. DOE has been used in the T&E of naval systems in the past. In a 2014 paper, McCarley and Jorris used DOE during the investigation of an F/A-18 E/F strafing anomaly. In their work, DOE was used as a means of gaining the most statistical information from the fewest number of test points and ultimately generated a predictive equation which explained the strafing anomaly [174].

The United States Naval Test Pilot School (USNTPS) teaches DOE as part of its short course, and this section is structured to follow the steps of the process [175]. In section 5.2.1 we detail the M&S environment, give a statement of the problem, and detail the scenario we will be modeling. In section 5.2.2 we describe the choice of experimental factors (variables) and detail how we measure them. In section 5.2.3 we discuss the measures of performance (MOP) for our experiment. Section III.D will detail how we plan to express the fused error distance as a function of the sensor errors.

5.2.1 M&S Environment and Statement of the Problem

As a truly autonomous system was not available for our evaluation, we elected to use a M&S environment for our research. Within the M&S environment we developed a scenario where an autonomous vehicle is reliant on its sensors to build its SA. The scenario was developed in such a way that the only factors effecting the SA of the vehicle are the accuracy of its sensors. Within the scenario the vehicle was required to make a decision, currently reserved for qualified pilots, based only on its degraded sensors.

For this experiment we used the Advanced Framework for Simulation, Integration and Modeling (AFSIM) environment. AFSIM is an engagement and mission level simulation environment written in C++ originally developed by Boeing and now managed by the Air Force Research Laboratory (AFRL). AFSIM was developed to address analysis capability shortcomings in existing legacy simulation environments as well as to provide an environment built with more modern programming paradigms in mind. AFSIM can simulate missions from subsurface to space and across multiple levels of model fidelity [176]. As AFSIM has been used by both the USN and United States Air Force (USAF) to inform acquisition decisions and model aircraft system behavior. We elected to use it to generate evidence that may lead to certification of autonomous systems to make a decision that is currently reserved for qualified pilot [177].

We proposed the following scenario for analyzing the effects of sensor error on the SA of an autonomous vehicle (and we programmed it into a M&S environment): An autonomous UAV (we refer to it as the Bucket Fighter) is operating over hostile territory. It is in a stationary orbit to provide Intelligence, Surveillance and Reconnaissance (ISR) information to ground forces. The information it provides is essential for the overall mission to be accomplish. However, the Bucket Fighter can be considered a High Value Airborne Asset (HVAA) that is unable to defend itself. As the platform is considered HVAA there is a set range it is required to maintain from threat aircraft. A fully qualified pilot is expected to take in the information available to them (both from communications with other assets and onboard systems) to determine when an aircraft reaches one of these pre-briefed limits. When a threat aircraft reaches a defined range, the Bucket Fighter will be required to RETROGRADE (withdraw from station in response to a threat, continue mission as able). Once the threat is no longer a factor, the vehicle can RESET to its orbit.

During a RETROGRADE, the ISR platform can continue to complete its assigned mission. When a threat aircraft reaches a defined range, the Bucket Fighter will be required to SCRAM (egress for defensive or survival reasons). If the UAV were to execute a SCRAM, it will no longer be able to provide support for ground forces, as a RESET is not authorized after a SCRAM. A description of these terms, and others used by the DoD, can be found in Reference [178].

For the sake of this hypothetical scenario we set the RETROGRADE and SCRAM ranges to 20 and 10 nautical miles (nm). An autonomous UAV's ability to accurately identity when a threat aircraft has reached its RETROGRADE and SCRAM range as critical for it to perform its mission. If it were to RETROGRADE or SCRAM to early it may lead to a unacceptable degradation to the assigned mission (ISR support for ground forces). If it were to RETROGRADE or SCRAM to late it may lead to a situation where a threat aircraft would engage the defenseless HVAA.

5.2.2 Experiment Factors (Variables)

Within the M&S environment, we installed two sensors on the Bucket Fighter (a generic InfraRed Search and Track (IRST) and a generic air-to-air radar). Both sensors were given an unlimited field of view and had the ability to track the threat aircraft for the duration of the simulation. In the M&S environment, we had the ability to add errors into each sensor in the form of a σ (Standard Deviation (SD)) value. These errors can be applied to the azimuth, elevation and range of the track.

Figure 5.4 is a pictorial of these the parameters. It is assumed that the only factors (environmental, mechanical or other) that can cause degradation to the individual sensor can be illustrated by the errors detailed above.

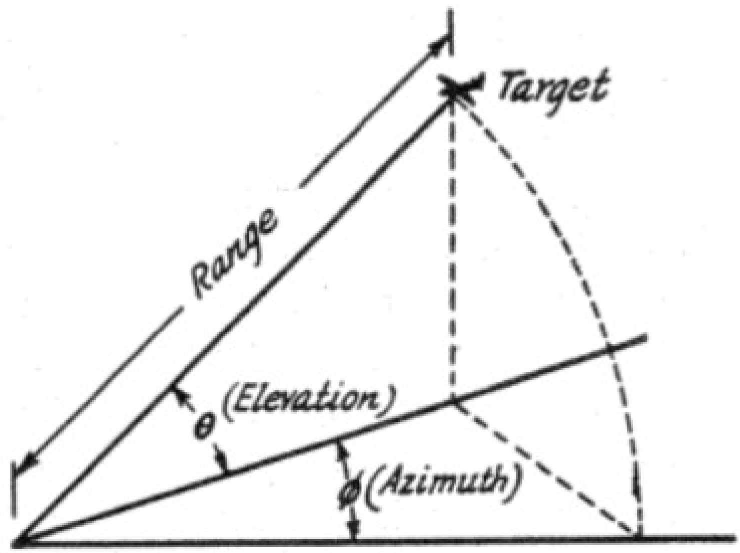

Figure 5.4: Graphical Depiction of the the Three Possible Error Parameters of the Sensors Installed on the Bucket Fighter [179].

During the scenario, the M&S package generates a random number to determine where on the normal distribution to pull the error value for each sensor. This error shifts each time the individual sensor performs a sweep. The errors are constant at each point in the simulation of the same scenario to enable repeatable results.

The Bucket Fighter had the ability to fuse the tracks provided by the IRST and radar. This fused track is based not only on the raw sensor data, but it uses velocity measurements and any past detection to build a predictable model for the track. This enables the autonomous UAV to more accurately track the target the longer it has been tracked by the sensors. Figure 5.5 contains two screen captures from a test run.

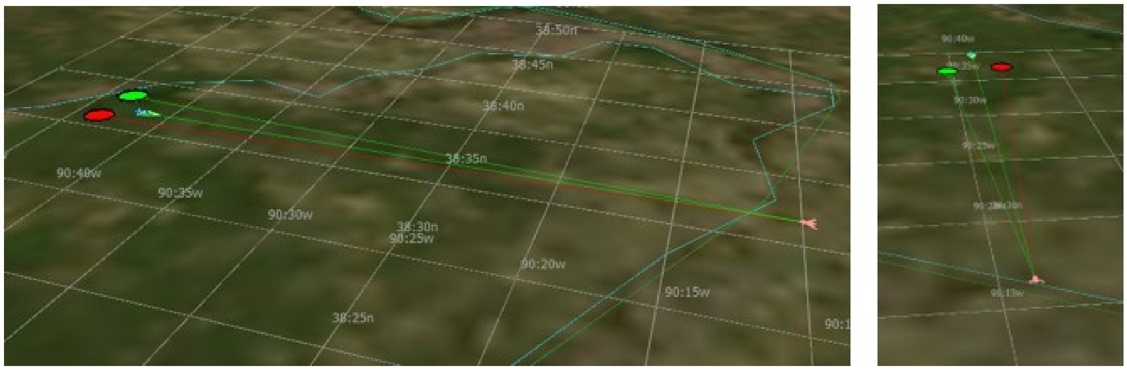

Figure 5.5: Two Screen Captures From the M&S Environment Depicting the Threat Location Based on the IRST (Red), Radar (Green), and Fused Track (White Triangle). The Threat Aircraft is Approximately 20 nm from the Bucket Fighter (UAV in the East). The Image to the Left is a View From the South and Slightly Elevated from the Engagement. The Image on the Right Depicts the Engagement From an Elevated Position in the East. The Error σ Values Were: IRST Azimuth: 9 Degrees, IRST Elevation: 3 Degrees, IRST Range: 9 nm, Radar Azimuth: 3 Degrees, Radar Elevation: 3 Degrees, Radar Range: 9 nm [180]

For DOE we chose the factors to be azimuth error, elevation error, and range error as resident in the radar and the IRST. This will give a total of six factors in the experiment with one level each (six variables). For each of the six factors, we use the following null hypothesis: No statistical significance can be found between the "error value" (IRST/radar azimuth, elevation, range) and the error distance (distance between the fused track and the threat aircraft).

5.2.3 Measures of Performance (MOP)

We are attempting to measure the SA provided to an autonomous system during periods of degraded sensor output. Therefore, we elected to use error distance as the Measures of Performance (MOP) in this research. In particular we measured the error distance at 20 and 10 nm (correspond to our hypothetical RETROGRADE and SCRAM range). Based on the errors inherent in the sensors (the six error σs), we hypothesized we could provide a predictive equation that would give the error distance at 20 and 10 nm. We use SME opinion (four senior naval officers who have extensive experience in dealing with RETROGRADE and SCRAM situations) to determine what error distance corresponds to inadequate SA to make a decision normally reserved for qualified pilots.

5.2.4 Fused Error Distance as a Function of Sensor Error

With the assistance of researchers from AFRL (Dayton, Ohio) and analysts from the Naval Air Warfare Center Aircraft Division (NAWCAD) (Patuxent River, Maryland) we adjusted a demonstration simulation from the standard unclassified AFSIM training software to meet the needs of our research. All output data from AFSIM used in this research was approved for public release [180].

We started with the Bucket Fighter providing ISR information to notional ground forces from a static location. We then elected to place a threat aircraft 60 nm from the Bucket Fighter. Both platforms were placed at 20,000 ft MSL and the threat aircraft tracked directly at the Bucket Fighter at 300 kts. For this hypothetical scenario, it can be assumed that the Bucket Fighter's only method of building an air picture (its SA of what is around it while airborne) is through its onboard sensors (a generic IRST and generic radar). Figure 5.6 is a screen capture depicting a top view from the start of the scenario with the Bucket Fighter in the

East, and the threat aircraft tracking inbound from the West. We studied the effect of sensor error on error distance (distance between the fused track and the actual location of the threat aircraft) at 20 and 10 nm in an attempt to quantify the SA level of the Bucket Fighter at critical decision points (RETROGRADE and SCRAM range) to determine at which point the SA provided to the Bucket Fighter was sufficient to make a sound aeronautical decision.

Figure 5.6: Screen Capture From the Start of a Test Run. The Threat Aircraft is in the West and the Bucket Fighter (UAV) is in the East [180].

Equation 5.1 is the multiple regression model that explains the relationship between Y (error distance/the independent variable) and multiple X_X values (the six error σs/dependent variables): X_1 = IRST azimuth σ value, X_2 = IRST elevation σ value, X_3 = IRST range σ value, X_4 = radar azimuth σ value, X_5 = radar elevation σ value, X_6 = radar range σ value. The corresponding β_x values are the relative weights of each variable and β_0 is the Y intercept. The ϵ term represents the error that exists within the model that cannot be accounted for and will drop out when we develop our predictive equation ($\hat{Y}$). Table 5.1 summarizes the various terms in

Term	Definition
Y	Error Delta/Independent Variable
X_1	IRST Azimuth σ Value
X_2	IRST Elevation σ Value
X_3	IRST Range σ Value
X_4	Radar Azimuth σ Value
X_5	Radar Elevation σ Value
X_6	Radar Range σ Value

Term	Definition
β_0	Y Intercept
β_1	Weight of the X_1 Variable
β_2	Weight of the X_2 Variable
β_3	Weight of the X_3 Variable
β_4	Weight of the X_4 Variable
β_5	Weight of the X_5 Variable
β_6	Weight of the X_6 Variable
ϵ	Error Within the Model

Table 5.1: Summary of the Terms in the Multiple Regression Model (Equation 5.1).

Equation 5.1.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_5 + \beta_6 X_6 + \epsilon \quad (5.1)$$

5.3 Experimental Results and Analysis

In Section 5.2 we developed a multiple regression model where we express the fused error distance as a function of the various sensor errors in the sensor network. In Section 5.3.1 we describe how we gathered data at various error σ levels to characterize the system. In Section 5.3.2 we then performed multiple variable regression analysis on the data gathered in the M&S environment to populate the variables in Equation 5.1 at 20 and 10 nm. In Section 5.2.3 we develop inequalities that define sufficient SA for an autonomous vehicle to make a decision that is currently reserved for qualified pilots.

5.3.1 Conduct of the Experiment

In an attempt to limit the scope of possible errors, and provide useful data to analyze, we limited the error σ to between three and seven (nm or degrees). We programmed in the ability to introduce three error variables into each of the sensors (azimuth (degrees), elevation (degrees) and range (nm)) in the form of defining one σ for each variable. For this research we varied the six variables between three, five and seven at the start of each test run and recorded the the observed error distance (distance between the fused track generated by the autonomous UAV and actual location of the threat aircraft) within the M&S environment. By manually updating the six σ values with all 729 combinations between each run, we hoped to provide enough data to generate predictive equations through multiple variable regression analysis. Equations 5.2 and 5.3 are the predictive equations (at 20 and 10 nm) we plan on population with the results of our regression analysis. Table 5.2 summarizes the various terms in Equations 5.2 and 5.3. The completed equations will be used to provide a quantitative evaluation of an autonomous systems SA to complete a task currently reserved for qualified pilots. Table 5.3 is a 10 run subset of the 729 combinations we plan on evaluating.

$$\hat{Y_{20}} = b_{0-20} + b_{1-20}X_1 + b_{2-20}X_2 + b_{3-20}X_3 + b_{4-20}X_4 + b_{5-20}X_5 + b_{6-20}X_6 \quad (5.2)$$

$$\hat{Y_{10}} = b_{0-10} + b_{1-10}X_1 + b_{2-10}X_2 + b_{3-10}X_3 + b_{4-10}X_4 + b_{5-10}X_5 + b_{6-10}X_6 \quad (5.3)$$

Term	Definition	Term	Definition
$\hat{Y_{20}}$	Predictive Error at 20 nm	b_{0-x}	Y Int. for the x Equ. (20/10)
$\hat{Y_{20}}$	Predictive Error at 10 nm	b_{1-x}	Weight of X_1, Equ. x (20/10)
X_1	IRST Azimuth σ Value	b_{2-x}	Weight of X_2, Equ. x (20/10)
X_2	IRST Elevation σ Value	b_{3-x}	Weight of X_3, Equ. x (20/10)
X_3	IRST Range σ Value	b_{4-x}	Weight of X_4, Equ. x (20/10)
X_4	Radar Azimuth σ Value	b_{5-x}	Weight of X_5, Equ. x (20/10)
X_5	Radar Elevation σ Value	b_{6-x}	Weight of X_6, Equ. x (20/10)
X_6	Radar Range σ Value		

Table 5.2: Summary of the Terms in the Predictive Equations (Equation 5.2 and 5.3).

Run #	Y_{20}	Y_{10}	X_1	X_2	X_3	X_4	X_5	X_6
50			3	3	5	7	5	5
141			3	5	7	3	5	7
248			5	3	3	3	5	5
339			5	5	3	5	5	7
397			5	5	7	7	3	3
469			5	7	7	5	3	3
554			7	3	7	5	5	5
594			7	5	3	7	7	7
656			7	7	3	3	7	5
723			7	7	7	7	3	7

Table 5.3: 10 of the 729 Data Points. Y_{20} is the 20 nm Error Distance in Meters. Y_{10} is the 10 nm Error Distance in Meters. X_1 is the Value of One σ Error in IRST Azimuth in Degrees. X_2 is the Value of One σ Error in IRST Elevation in Degrees. X_3 is the Value of One σ Error in IRST Range in nm. X_4 is the Value of One σ Error in Radar Azimuth in Degrees. X_5 is the Value of One σ Error in Radar Elevation in Degrees. X_6 is the Value of One σ Error in Radar Range in nm.

Run #	Y_{20}	Y_{10}	X_1	X_2	X_3	X_4	X_5	X_6
50	467.9	342.9	3	3	5	7	5	5
141	325.8	181.5	3	5	7	3	5	7
248	331.2	243.3	5	3	3	3	5	5
339	496.7	381.7	5	5	3	5	5	7
397	617.9	481.6	5	5	7	7	3	3
469	434.0	320.5	5	7	7	5	3	3
554	567.3	412.1	7	3	7	5	5	5
594	818.1	600.9	7	5	3	7	7	7
656	932.4	765.5	7	7	3	3	7	5
723	813.4	615	7	7	7	7	3	7

Table 5.4: 10 of the 729 Data Points. Y_{20} is the 20 nm Error Distance in Meters. Y_{10} is the 10 nm Error Distance in Meters. X_1 is the Value of One σ Error in IRST Azimuth in Degrees. X_2 is the Value of One σ Error in IRST Elevation in Degrees. X_3 is the Value of One σ Error in IRST Range in nm. X_4 is the Value of One σ Error in Radar Azimuth in Degrees. X_5 is the Value of One σ Error in Radar Elevation in Degrees. X_6 is the Value of One σ Error in Radar Range in nm.

5.3.2 Analysis of the Data

As discussed in Section 5.3.1, we planned on evaluating 729 different combinations of the six variables (error σs). Table 5.4 is a subset of 10 (the same 10 as Table 5.3) of the runs with the observed error distance (measured in meters) at 20 and 10 nm. All 729 simulations can be found in Table A.2.

We then used multiple variable regression analysis resident in Microsoft Excel to preform regression analysis on the 729 data points to determine the effects each independent variable (the six σ values) had on the two dependent variables (error distance at 20 and 10 nm). The 20 nm data adjusted R-Squared Value (indicates the percentage of the variance in the dependent variable that the independent variables explain collectively) was 0.822, and the 10 nm adjusted R-Squared Value was 0.818. R-Squared describes levels of predictive accuracy with 0.75, 0.50, 0.25,

20 Mile Regression Data				10 Mile Regression Data		
$\hat{Y_{20}}$ b Term	Coefficient	P-Value		$\hat{Y_{10}}$ b Term	Coefficient	P-Value
b_{0-20}	-528.337	4.02E-80		b_{0-10}	-441.693	5.90E-73
b_{1-20}	57.427	3.80E-123		b_{1-10}	49.665	9.80E-119
b_{2-20}	54.105	2.31E-113		b_{2-10}	46.854	2.10E-109
b_{3-20}	4.653	1.91E-02		b_{3-10}	-4.607	9.01E-03
b_{4-20}	54.649	5.75E-115		b_{4-10}	47.578	8.30E-112
b_{5-20}	61.384	9.58E-135		b_{5-10}	53.085	4.70E-130
b_{6-20}	-10.208	3.31E-07		b_{6-10}	-15.638	4.84E-18

Table 5.5: 20 and 10 nm Regression Data Obtained Through Microsoft Excel Multiple Regression Analysis.

respectively, describing substantial, moderate, or weak [181]. The Analysis of Variation (ANOVA) Significance F value was 6.831E-267 for 20, and 8.674E-263 for 10 nm (both of which show an extremely high statistical significance for the respective model). Table 5.5 details the relative coefficients for the predictive equation and the individual P-Values. All of the P-values are well below 0.05. Therefore, we must reject the 6 null hypotheses as there is a significant relationship between each sensor error value and the fused track error distance.

Equations 5.4 and 5.5 are predictive equations ($\hat{Y}$) that depict an anticipated fused error distance (dependent variable) based on the the various error σs (independent variables) internal to the system at 20 and 10 nm respectively (X_1 = IRST azimuth σ value, X_2 = IRST elevation σ value, X_3 = IRST range σ value, X_4 = radar azimuth σ value, X_5 = radar elevation σ value, X_6 = radar range σ value). The corresponding b_X values are the relative weights of each variable and b_0 is the Y intercept from Table 5.5.

$$\hat{Y_{20}} = -528.337 + 57.427X_1 + 54.105X_2 + 4.653X_3 + 54.649X_4 + 61.384X_5 - 10.208X_6 \tag{5.4}$$

$$\hat{Y_{10}} = -441.693 + 49.665X_1 + 46.854X_2 - 4.607X_3 + 47.578X_4 + 53.085X_5 - 15.638X_6 \tag{5.5}$$

Next, we used a random number generator (integers between three and seven) to populated 25 test points for the evaluation of the predictive equations. We elected to limit out evaluation of the regression analysis to σs between three and seven, as that was the population of the data that we used for the regression analysis. Table 5.6 details these test points and their error observed distances at 20 and 10 nm. Table 5.7 then compares the predicted error distance and observed error distance from the M&S environment. We elected to use the absolute error vice the actual error as the actual error has a tendency to reduce the average error across multiple data points. The predicative equations generated error distances across the 25 points with less then a 10% average error at both 20 and 10 nm (distance between the observed range and fused track). While some of the errors seem extreme (in excess of 20% in some cases, these are the result of σ errors in the range of the sensor in excess of 5 nm. Under normal operations, errors of this magnitude would be highly unlikely in a fielded system. In addition, all of the deltas that were in excess of 10% were reflective of values that had the predictive equation necessitating a RETROGRADE or SCRAM before the threat aircraft actually reached the RETROGRADE or SCRAM range. Therefore, while the SA provided by the system would cause the Bucket Fighter to depart station prior to its requirement, the decision would be safer than if error were in the opposite direction (having the Bucket Fighter remain on station past RETROGRADE or SCRAM range).

5.3.3 DOE Conclusions

Based on this output and SME (four senior naval officers who have extensive experience in dealing with RETROGRADE and SCRAM situations) opinion, we determined that if the system could generate a error distance less than 800 meters at 20 nm, and 400 meters at 10 nm, then the SA provided by its sensors is accurate enough for it to make the RETROGRADE or SCRAM decision normally reserved for qualified pilots. As the error distance from the predictive equation is within 10% of the observed error distance we used 727 m for 20 nm (worst case: $727+(727*.1) = 799.6$), and 363 for 10 nm (worst case: $363+(363*.1) = 399.3$). Equations 5.4 and 5.5 where then translated to be inequalities, Equations 5.6 and 5.7. When Equation 5.6 is true, the SA provided by the onboard sensors is sufficient to make a sound RETROGRADE decision at 20 nm. When Equation 5.7 is true, the SA provided by the onboard sensors is sufficient to make a sound SCRAM decision at 10 nm. If Equation 5.6 or 5.7 were to be false, the SA provided by the onboard sensors is not adequate for making a sound RETROGRADE or SCRAM decision.

Test Run #	Y_{20}	Y_{10}	X_1	X_2	X_3	X_4	X_5	X_6
T - 1	476.5	362.9	7	5	4	3	3	4
T - 2	588.5	458.0	5	6	4	6	4	5
T - 3	537.4	411.0	5	5	4	6	5	6
T - 4	307.0	195.0	3	5	5	4	4	6
T - 5	451.1	338.6	4	5	4	6	4	5
T - 6	396.5	271.7	7	3	6	3	4	5
T - 7	517.5	384.6	3	5	5	4	7	5
T - 8	481.0	367.9	5	5	3	4	5	7
T - 9	291.6	210.8	5	3	4	3	4	3
T - 10	493.6	387.0	6	6	4	4	4	3
T - 11	394.0	291.6	6	4	4	3	4	4
T - 12	859.2	687.7	6	7	5	5	7	4
T - 13	750.7	571.3	3	7	7	5	7	5
T - 14	824.5	645.3	7	7	6	5	6	5
T - 15	628.6	463.5	7	3	6	5	6	6
T - 16	470.0	345.3	4	5	5	6	4	5
T - 17	425.8	289.8	5	3	7	6	3	5
T - 18	463.8	322.0	5	3	5	6	6	7
T - 19	742.9	586.2	4	7	7	4	7	3
T - 20	571.8	400.6	7	3	6	4	6	7
T - 21	874.1	710.4	7	6	3	7	5	7
T - 22	394.7	279.7	3	6	4	5	3	7
T - 23	540.5	418.4	7	4	5	6	3	3
T - 24	471.1	345.9	5	6	5	4	4	5
T - 25	384.8	262.7	5	5	5	3	4	6

Table 5.6: Results From 25 Test Runs of Randomly Generated σ Values. Y_{20} is the 20 nm Error Distance in Meters. Y_{10} is the 10 mn Error Distance in Meters. X_1 is the Value of One σ Error in IRST Azimuth in Degrees. X_2 is the Value of One σ Error in IRST Elevation in Degrees. X_3 is the Value of One σ Error in IRST Range in nm. X_4 is the Value of One σ Error in Radar Azimuth in Degrees. X_5 is the Value of One σ Error in Radar Elevation in Degrees. X_6 is the Value of One σ Error in Radar Range in nm.

Run #	Y_{20}	$\hat{Y_{20}}$	Delta (m/%)
T - 1	476.5	460.1	16.4/3.44%
T - 2	588.5	614.5	26.0/4.41%
T - 3	537.4	611.5	74.1/13.80%
T - 4	307.0	327.0	20.0/6.52%
T - 5	451.1	502.9	51.8/11.49%
T - 6	396.5	405.1	8.6/2.16%
T - 7	517.5	521.4	3.9/0.75%
T - 8	481.0	491.0	10.0/2.09%
T - 9	291.6	308.6	17.0/5.84%
T - 10	493.6	583.0	89.4/18.11%
T - 11	394.0	409.9	15.9/4.05%
T - 12	859.2	866.7	7.5/0.87%
T - 13	750.7	686.2	64.5/8.59%
T - 14	824.5	853.5	29.0/3.52%
T - 15	628.6	626.9	1.7/0.27%
T - 16	470.0	503.9	33.9/7.22%
T - 17	425.8	393.8	32.0/7.52%
T - 18	463.8	555.5	91.7/19.77%
T - 19	742.9	709.4	33.5/4.51%
T - 20	571.8	562.1	9.7/1.70%
T - 21	874.1	823.9	50.2/5.74%
T - 22	394.7	363.2	31.5/7.99%
T - 23	540.5	581.1	40.6/7.52%
T - 24	471.1	506.2	35.1/7.44%
T - 25	384.8	387.2	2.4/0.63%
Average Error			**6.24%**

Run #	Y_{10}	$\hat{Y_{10}}$	Delta (m/%)
T - 1	362.9	361.2	1.7/0.46%
T - 2	458.0	488.9	30.9/6.76%
T - 3	411.0	479.5	68.5/16.68%
T - 4	195.0	227.4	32.4/16.60%
T - 5	338.6	392.4	53.8/15.90%
T - 6	271.7	295.8	24.1/8.86%
T - 7	384.6	402.3	17.7/4.59%
T - 8	367.9	373.4	5.5/1.48%
T - 9	210.8	236.9	26.1/12.39%
T - 10	387.0	474.7	87.7/22.67%
T - 11	291.6	317.8	26.2/8.99%
T - 12	687.7	708.2	20.5/2.98%
T - 13	571.3	534.3	37.0/6.47%
T - 14	645.3	684.5	39.2/6.08%
T - 15	463.5	481.5	18.0/3.87%
T - 16	345.3	387.8	42.5/12.31%
T - 17	289.8	281.5	8.3/2.87%
T - 18	322.0	418.7	96.7/30.02%
T - 19	586.2	567.7	18.5/3.16%
T - 20	400.6	418.2	17.6/4.40%
T - 21	710.4	662.3	48.1/6.78%
T - 22	279.7	257.7	22.0/7.87%
T - 23	418.4	468.1	49.7/11.89%
T - 24	345.9	389.2	43.3/12.51%
T - 25	262.7	279.1	16.4/6.25%
Average Error			**9.31%**

Table 5.7: Results From 25 Test Runs Y_x, the Corresponding Results From of Predictive Equations $\hat{Y_x}$, the Absolute Distance Between the Two in Meters and Percentage. The Error Percentages are also Summarized.

$$727 > -528.337 + 57.427X_1 + 54.105X_2 + 4.653X_3 + 54.649X_4 + 61.384X_5 + -10.208X_6 \tag{5.6}$$

$$363 > -441.693 + 49.665X_1 + 46.854X_2 - 4.607X_3 + 47.578X_4 + 53.085X_5 - 15.638X_6 \tag{5.7}$$

5.4 Chapter Conclusion

In this chapter, we demonstrated that a relationship (objective measure) can be defined for autonomous vehicle SA (subjective end) and sensor degradation. Section 5.1 details how defining a objective measure for a subjective end is not a new idea within the flight test community and highlighted inadequate vehicle SA in an autonomous technology demonstration vehicle. Section 5.2 and 5.3 dealt with M&S of a hypothetical simplified sensor network to define the relationship. Future work that focused on defining this relationship on a mature system during flight test would give vehicle designers the ability to program a vehicle to complete tasks currently reserved for qualified pilots under off nominal conditions and eventually obtain a SOF certification.

Chapter 6: Conclusions

6.1 Summary

This dissertation was prepared in close coordination with naval SOF clearance officials to determine a path forward for certifying autonomy in naval aircraft. A method for certifying a vehicle to make decisions when a qualified pilot/operator is not in the loop does not currently exist. We proposed, and certification officials concurred, the following methodology as a possible avenue for certifying autonomy in the hopes that lessons learned from its exercise will help develop an approved process before the first autonomous system is acquired by the Navy:

1. Define the requirements (normally reserved for a pilot) to execute autonomous behavior. These requirements must be developed through coordination with SOF certification officials, the naval T&E community, and fleet officials who currently certify pilots as fully qualified. A specification will then be developed that can be used to verify the requirements have been completely and accurately specified.

2. Develop the clearance envelope where the system will be allowed to exhibit non-deterministic behavior (the exact behavior of the system cannot be de-

termined based upon the input conditions). If the system were to encounter the edge of this envelope it would revert to deterministic behavior (based on known input conditions, the vehicle will exhibit a known behavior).

3. Analyze the specification to ensure the requirements of the system are met.

4. Develop a protocol/set of control laws with traceability to the verified specification. This way formal methods will satisfy the requirements of the system, as the protocol/control laws will have formally verified properties.

5. Limited M&S of the algorithms/control laws as a risk reduction tool prior to flight test. This will attempt to show the system will display non-deterministic behavior only while it is within the clearance envelope.

6. Design the process for flight test. Most conventional flight test techniques are designed for a pilot to test an unproven system. In this case, test points will need to be developed that demonstrate under controlled (DT) and operationally relevant conditions (OT) the system under test can complete the assigned mission.

7. Execute DT and OT on the autonomous system under test.

8. Full report of the tests conducted on the system under test.

Chapter 3 (which was derived from Reference [76]) details the execution of the first 4 steps of the proposed methodology. First, we defined the requirements for an autonomous vehicle to land a large rotercraft in an unprepared LZ and developed a specification. Then, we developed a clearance envelope by using an established procedure pilots currently execute to accomplish the mission. Next, we verified the specification. Finally a proposed protocol was developed based on the verified specification. Naval SOF certification officials were satisfied, and requested that the research continue. They requested an evaluation of the methodology using the technology demonstration vehicle developed by AFS.

Chapter 4 (which was derived from Reference [144]) covered the final three steps of the proposed methodology (the missing step was performed by AFS prior to flight test [132]). The system under test was able to demonstrate it could accomplish the CAL/LZ mission under controlled conditions. However, as outlined in Section 4.4, the system lacked the required SA of its environment to complete the task currently reserved for qualified pilots under mission relevant conditions when off nominal conditions were encountered.

AFS demonstrated that an autonomous vehicle was capable of sensing its environment, using that information to build onboard SA, and making appropriate aeronautical decisions (for a task currently reserved for fully qualified pilots) based on that SA. As the demonstrator was between a TRL 4 and 5, it was never designed to gain a SOF certification when operated autonomously. The obstacle threshold is just one example that would prevent the vehicle from advancing beyond TRL 7. AFS was given a requirement to avoid obstacles that would cause a hazard during landing. It defined an 11 in obstacle as a hazard, vice identifying what that obstacle was, then it ensured the vehicle would not land near the 11 in hazard. This gave the vehicle inadequate SA in mission representative environments.

However, evaluating the methodology highlighted a major issue. If a vehicle uses its sensors to build its SA, there should be a point where the vehicle SA will drop to an unsatisfactory level for making sound aeronautical decisions as the sensor output degrades. If that point can be defined objectively it could be given to vehicle designers, and certification officials, to be programmed into the clearance envelope outlined in step 2 of the methodology. Chapter 5 (which was derived from Reference [182]) focused on defining this point in a M&S environment through modeling errors within a simple sensor network and can be seen as exercising step 5 of the proposed methodology. Despite being TRL 3 or 4, Sections 5.2 and 5.3 demonstrated that there could be a relationship between sensor degradation and obtaining adequate vehicle SA to complete a task currently reserved for qualified pilots. Before a SOF clearance would be granted, both OT and DT would be required to ensure that M&S findings translated to real world results.

6.2 Original Contributions

As I am currently a senior naval officer with contacts and established relationships within the naval acquisitions community (to include T&E Community Leadership, SOF certification officials, and the NAWCAD center for autonomy) this research was given many unique opportunities not normally afforded to University studies. These opportunities included access to senior officials for interviews and guidance, access to existing data sets within the Navy, and access to DoD approved M&S environments.

The original contribution to knowledge contained in this dissertation include:

- Proposed methodology for obtaining a naval aviation SOF certification allowing a decision engine to complete a task currently reserved for a qualified pilot. Then the exercise of the methodology to help build a path forward for certifying autonomy. Currently the United States Navy (USN) does not have a path forward for certifying autonomy. This contribution will influence future certification standards and procedures for this emerging requirement.

- Definition of the requirements a decision engine must complete if it were to be approved to complete the CAL/LZ mission autonomously (a task currently reserved for a qualified pilot). Then use of a formal methods approach to ensure the actions taken by a developed protocol will satisfy the requirements defined. This contribution exercised the first four steps of the methodology proposed in Section 1.4 and provide artifacts to certification officials for a possible SOF clearance allowing a decision engine to complete a mission currently reserved for a qualified pilot.

- Development of flight test matrices (one for DT and one for OT) for an autonomous vehicle to complete the CAL/LZ mission. Followed by analysis of both DT and OT flight test data of an autonomous vehicle completing a task currently reserved for a qualified pilot (CAL/LZ mission). This contribution exercised the last three steps of the methodology proposed in Section 1.4 and provided artifacts to certification officials for a possible SOF clearance allowing a decision engine to complete a mission currently reserved for a qualified pilot.

- Development of an objective measure for autonomous vehicle SA that accounted for sensor degradation within a Department of Defense (DoD) recognized M&S environment. The measure specifically evaluated the effects of sensor degradation on error distance of a fused track of a threat aircraft. We used Design of Experiments (DOE) to determine the effects of sensor degradation and produce a set of predictive equations for the error distance of the fused track. Then used Subject Matter Expert (SME) opinion to define the point at which (within this scenario) the fused error distance was inadequate to make a decision currently reserved for a qualified pilot. This contribution exercised the fifth step of the methodology proposed in Section 1.4 and provided results that if confirmed during flight test could have lead to a SOF clearance allowing a decision engine to complete a mission currently reserved for a qualified pilot.

6.3 Outlook for Future Work

This subject area offers multiple avenues for future work. The formal methods approach outlined in Chapter 3 would require an extensive amount of research to fully flush out the possible scenarios an autonomous vehicle would need to negotiate before a SOF certification could be obtained.

Flight test has been identified as one of the major issues in the V&V process for autonomy. The current paradigm is to test all scenarios to see how software will react. Without a human in the loop, there are a limitless number of test points for full, comprehensive test. Defining the cut off line within the decision space is a risk decision that will need to be made in the near future if we are to have true autonomy in aviation (military or civilian).

M&S has been a benchmark of research for decades. However, unless the model has the fidelity required by the DoD it will have limited use in the acquisition process. Future work in M&S in support of autonomy in military systems will most likely be classified based on the environments used. While they may lead to autonomous systems certifications within the DoD, there will be limited publishable material.

The proposed methodology developed for this dissertation is just one possible path for certifying autonomy. It was developed in coordination with naval flight clearance officials. Future work that develops possible paths forward for certifying autonomy should include the involvement of certification officials. In this work, we exercised each step of our proposed methodology (but not with the same system under test). Future work that uses this process with different autonomous systems, or uses this process from the beginning through certification of one system, would develop more lessons learned for certifying autonomy. Ultimately an approved process needs to be in place before we can certify an autonomous system to make decisions currently reserved for qualified pilots.

Appendix A: Experimental Results

The appendix contains the complete data sets that were summarized in various Chapters of this dissertation. Section A.1 contains the 256 possible outcomes of the 8 protocol assessments truncated in 3.2. Section Table A.2 contains the results of the 729 AFSIM simulation runs truncated in Table 5.4.

A.1 All Possible Outcomes of the Eight Protocol Assessments Truncated in Table 3.2

Table A.1 contains the 256 possible outcomes of the eight protocol assessments truncated in Table 3.2.

Table A.1: 256 Possible Outcomes of the 8 Protocol Assessments Truncated in Table 3.2

Outcome #	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	1

Continued on next page

Table A.1 – continued from previous page

Outcome #	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8
3	0	0	0	0	0	0	1	0
4	0	0	0	0	0	0	1	1
5	0	0	0	0	0	1	0	0
6	0	0	0	0	0	1	0	1
7	0	0	0	0	0	1	1	0
8	0	0	0	0	0	1	1	1
9	0	0	0	0	1	0	0	0
10	0	0	0	0	1	0	0	1
11	0	0	0	0	1	0	1	0
12	0	0	0	0	1	0	1	1
13	0	0	0	0	1	1	0	0
14	0	0	0	0	1	1	0	1
15	0	0	0	0	1	1	1	0
16	0	0	0	0	1	1	1	1
17	0	0	0	1	0	0	0	0
18	0	0	0	1	0	0	0	1
19	0	0	0	1	0	0	1	0
20	0	0	0	1	0	0	1	1
21	0	0	0	1	0	1	0	0

Continued on next page

Table A.1 – continued from previous page

Outcome #	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8
22	0	0	0	1	0	1	0	1
23	0	0	0	1	0	1	1	0
24	0	0	0	1	0	1	1	1
25	0	0	0	1	1	0	0	0
26	0	0	0	1	1	0	0	1
27	0	0	0	1	1	0	1	0
28	0	0	0	1	1	0	1	1
29	0	0	0	1	1	1	0	0
30	0	0	0	1	1	1	0	1
31	0	0	0	1	1	1	1	0
32	0	0	0	1	1	1	1	1
33	0	0	1	0	0	0	0	0
34	0	0	1	0	0	0	0	1
35	0	0	1	0	0	0	1	0
36	0	0	1	0	0	0	1	1
37	0	0	1	0	0	1	0	0
38	0	0	1	0	0	1	0	1
39	0	0	1	0	0	1	1	0
40	0	0	1	0	0	1	1	1

Continued on next page

Table A.1 – continued from previous page

Outcome #	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8
41	0	0	1	0	1	0	0	0
42	0	0	1	0	1	0	0	1
43	0	0	1	0	1	0	1	0
44	0	0	1	0	1	0	1	1
45	0	0	1	0	1	1	0	0
46	0	0	1	0	1	1	0	1
47	0	0	1	0	1	1	1	0
48	0	0	1	0	1	1	1	1
49	0	0	1	1	0	0	0	0
50	0	0	1	1	0	0	0	1
51	0	0	1	1	0	0	1	0
52	0	0	1	1	0	0	1	1
53	0	0	1	1	0	1	0	0
54	0	0	1	1	0	1	0	1
55	0	0	1	1	0	1	1	0
56	0	0	1	1	0	1	1	1
57	0	0	1	1	1	0	0	0
58	0	0	1	1	1	0	0	1
59	0	0	1	1	1	0	1	0

Continued on next page

Table A.1 – continued from previous page

Outcome #	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8
60	0	0	1	1	1	0	1	1
61	0	0	1	1	1	1	0	0
62	0	0	1	1	1	1	0	1
63	0	0	1	1	1	1	1	0
64	0	0	1	1	1	1	1	1
65	0	1	0	0	0	0	0	0
66	0	1	0	0	0	0	0	1
67	0	1	0	0	0	0	1	0
68	0	1	0	0	0	0	1	1
69	0	1	0	0	0	1	0	0
70	0	1	0	0	0	1	0	1
71	0	1	0	0	0	1	1	0
72	0	1	0	0	0	1	1	1
73	0	1	0	0	1	0	0	0
74	0	1	0	0	1	0	0	1
75	0	1	0	0	1	0	1	0
76	0	1	0	0	1	0	1	1
77	0	1	0	0	1	1	0	0
78	0	1	0	0	1	1	0	1

Continued on next page

Table A.1 – continued from previous page

Outcome #	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8
79	0	1	0	0	1	1	1	0
80	0	1	0	0	1	1	1	1
81	0	1	0	1	0	0	0	0
82	0	1	0	1	0	0	0	1
83	0	1	0	1	0	0	1	0
84	0	1	0	1	0	0	1	1
85	0	1	0	1	0	1	0	0
86	0	1	0	1	0	1	0	1
87	0	1	0	1	0	1	1	0
88	0	1	0	1	0	1	1	1
89	0	1	0	1	1	0	0	0
90	0	1	0	1	1	0	0	1
91	0	1	0	1	1	0	1	0
92	0	1	0	1	1	0	1	1
93	0	1	0	1	1	1	0	0
94	0	1	0	1	1	1	0	1
95	0	1	0	1	1	1	1	0
96	0	1	0	1	1	1	1	1
97	0	1	1	0	0	0	0	0

Continued on next page

Table A.1 – continued from previous page

Outcome #	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8
98	0	1	1	0	0	0	0	1
99	0	1	1	0	0	0	1	0
100	0	1	1	0	0	0	1	1
101	0	1	1	0	0	1	0	0
102	0	1	1	0	0	1	0	1
103	0	1	1	0	0	1	1	0
104	0	1	1	0	0	1	1	1
105	0	1	1	0	1	0	0	0
106	0	1	1	0	1	0	0	1
107	0	1	1	0	1	0	1	0
108	0	1	1	0	1	0	1	1
109	0	1	1	0	1	1	0	0
110	0	1	1	0	1	1	0	1
111	0	1	1	0	1	1	1	0
112	0	1	1	0	1	1	1	1
113	0	1	1	1	0	0	0	0
114	0	1	1	1	0	0	0	1
115	0	1	1	1	0	0	1	0
116	0	1	1	1	0	0	1	1

Continued on next page

Table A.1 – continued from previous page

Outcome #	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8
117	0	1	1	1	0	1	0	0
118	0	1	1	1	0	1	0	1
119	0	1	1	1	0	1	1	0
120	0	1	1	1	0	1	1	1
121	0	1	1	1	1	0	0	0
122	0	1	1	1	1	0	0	1
123	0	1	1	1	1	0	1	0
124	0	1	1	1	1	0	1	1
125	0	1	1	1	1	1	0	0
126	0	1	1	1	1	1	0	1
127	0	1	1	1	1	1	1	0
128	0	1	1	1	1	1	1	1
129	1	0	0	0	0	0	0	0
130	1	0	0	0	0	0	0	1
131	1	0	0	0	0	0	1	0
132	1	0	0	0	0	0	1	1
133	1	0	0	0	0	1	0	0
134	1	0	0	0	0	1	0	1
135	1	0	0	0	0	1	1	0

Continued on next page

Table A.1 – continued from previous page

Outcome #	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8
136	1	0	0	0	0	1	1	1
137	1	0	0	0	1	0	0	0
138	1	0	0	0	1	0	0	1
139	1	0	0	0	1	0	1	0
140	1	0	0	0	1	0	1	1
141	1	0	0	0	1	1	0	0
142	1	0	0	0	1	1	0	1
143	1	0	0	0	1	1	1	0
144	1	0	0	0	1	1	1	1
145	1	0	0	1	0	0	0	0
146	1	0	0	1	0	0	0	1
147	1	0	0	1	0	0	1	0
148	1	0	0	1	0	0	1	1
149	1	0	0	1	0	1	0	0
150	1	0	0	1	0	1	0	1
151	1	0	0	1	0	1	1	0
152	1	0	0	1	0	1	1	1
153	1	0	0	1	1	0	0	0
154	1	0	0	1	1	0	0	1

Continued on next page

Table A.1 – continued from previous page

Outcome #	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8
155	1	0	0	1	1	0	1	0
156	1	0	0	1	1	0	1	1
157	1	0	0	1	1	1	0	0
158	1	0	0	1	1	1	0	1
159	1	0	0	1	1	1	1	0
160	1	0	0	1	1	1	1	1
161	1	0	1	0	0	0	0	0
162	1	0	1	0	0	0	0	1
163	1	0	1	0	0	0	1	0
164	1	0	1	0	0	0	1	1
165	1	0	1	0	0	1	0	0
166	1	0	1	0	0	1	0	1
167	1	0	1	0	0	1	1	0
168	1	0	1	0	0	1	1	1
169	1	0	1	0	1	0	0	0
170	1	0	1	0	1	0	0	1
171	1	0	1	0	1	0	1	0
172	1	0	1	0	1	0	1	1
173	1	0	1	0	1	1	0	0

Continued on next page

Table A.1 – continued from previous page

Outcome #	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8
174	1	0	1	0	1	1	0	1
175	1	0	1	0	1	1	1	0
176	1	0	1	0	1	1	1	1
177	1	0	1	1	0	0	0	0
178	1	0	1	1	0	0	0	1
179	1	0	1	1	0	0	1	0
180	1	0	1	1	0	0	1	1
181	1	0	1	1	0	1	0	0
182	1	0	1	1	0	1	0	1
183	1	0	1	1	0	1	1	0
184	1	0	1	1	0	1	1	1
185	1	0	1	1	1	0	0	0
186	1	0	1	1	1	0	0	1
187	1	0	1	1	1	0	1	0
188	1	0	1	1	1	0	1	1
189	1	0	1	1	1	1	0	0
190	1	0	1	1	1	1	0	1
191	1	0	1	1	1	1	1	0
192	1	0	1	1	1	1	1	1

Continued on next page

Table A.1 – continued from previous page

Outcome #	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8
193	1	1	0	0	0	0	0	0
194	1	1	0	0	0	0	0	1
195	1	1	0	0	0	0	1	0
196	1	1	0	0	0	0	1	1
197	1	1	0	0	0	1	0	0
198	1	1	0	0	0	1	0	1
199	1	1	0	0	0	1	1	0
200	1	1	0	0	0	1	1	1
201	1	1	0	0	1	0	0	0
202	1	1	0	0	1	0	0	1
203	1	1	0	0	1	0	1	0
204	1	1	0	0	1	0	1	1
205	1	1	0	0	1	1	0	0
206	1	1	0	0	1	1	0	1
207	1	1	0	0	1	1	1	0
208	1	1	0	0	1	1	1	1
209	1	1	0	1	0	0	0	0
210	1	1	0	1	0	0	0	1
211	1	1	0	1	0	0	1	0

Continued on next page

Table A.1 – continued from previous page

Outcome #	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8
212	1	1	0	1	0	0	1	1
213	1	1	0	1	0	1	0	0
214	1	1	0	1	0	1	0	1
215	1	1	0	1	0	1	1	0
216	1	1	0	1	0	1	1	1
217	1	1	0	1	1	0	0	0
218	1	1	0	1	1	0	0	1
219	1	1	0	1	1	0	1	0
220	1	1	0	1	1	0	1	1
221	1	1	0	1	1	1	0	0
222	1	1	0	1	1	1	0	1
223	1	1	0	1	1	1	1	0
224	1	1	0	1	1	1	1	1
225	1	1	1	0	0	0	0	0
226	1	1	1	0	0	0	0	1
227	1	1	1	0	0	0	1	0
228	1	1	1	0	0	0	1	1
229	1	1	1	0	0	1	0	0
230	1	1	1	0	0	1	0	1

Continued on next page

Table A.1 – continued from previous page

Outcome #	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8
231	1	1	1	0	0	1	1	0
232	1	1	1	0	0	1	1	1
233	1	1	1	0	1	0	0	0
234	1	1	1	0	1	0	0	1
235	1	1	1	0	1	0	1	0
236	1	1	1	0	1	0	1	1
237	1	1	1	0	1	1	0	0
238	1	1	1	0	1	1	0	1
239	1	1	1	0	1	1	1	0
240	1	1	1	0	1	1	1	1
241	1	1	1	1	0	0	0	0
242	1	1	1	1	0	0	0	1
243	1	1	1	1	0	0	1	0
244	1	1	1	1	0	0	1	1
245	1	1	1	1	0	1	0	0
246	1	1	1	1	0	1	0	1
247	1	1	1	1	0	1	1	0
248	1	1	1	1	0	1	1	1
249	1	1	1	1	1	0	0	0

Continued on next page

Table A.1 – continued from previous page

Outcome #	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8
250	1	1	1	1	1	0	0	1
251	1	1	1	1	1	0	1	0
252	1	1	1	1	1	0	1	1
253	1	1	1	1	1	1	0	0
254	1	1	1	1	1	1	0	1
255	1	1	1	1	1	1	1	0
256	1	1	1	1	1	1	1	1

A.2 All 729 Combinations of the Six Variables (error σs) Truncated in Table 5.4

Table A.2 contains the 729 combinations of the six variables (error σs) truncated in Table 5.4.

Table A.2: Results of the 729 Simulations Truncated in Table 5.4. Y_{20} is the 20 nm Error Distance in Meters. Y_{10} is the 10 nm Error Distance in Meters. X_1 is the Value of One σ Error in IRST Azimuth in Degrees. X_2 is the Value of One σ Error in IRST Elevation in Degrees. X_3 is the Value of One σ Error in IRST Range in nm. X_4 is the Value of One σ Error in Radar Azimuth in Degrees. X_5 is the Value of One σ error in Radar Elevation in Degrees. X_6 is the Value of One σ Error in Radar Range in nm [180].

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
1	164.5	108.1	3	3	3	3	3	3
2	143.2	82.5	3	3	3	3	3	5
3	134.4	66.1	3	3	3	3	3	7
4	206.8	186.6	3	3	3	3	5	3
5	194.4	126.5	3	3	3	3	5	5
6	164.8	92.1	3	3	3	3	5	7
7	405	309	3	3	3	3	7	3
8	271	191.9	3	3	3	3	7	5
9	1096.6	896.2	3	3	3	3	7	7

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
10	253	182.3	3	3	3	5	3	3
11	190.4	123.6	3	3	3	5	3	5
12	161.8	90.5	3	3	3	5	3	7
13	349.1	262.9	3	3	3	5	5	3
14	241.5	167.6	3	3	3	5	5	5
15	192.2	116.5	3	3	3	5	5	7
16	493.1	385.8	3	3	3	5	7	3
17	317.9	233	3	3	3	5	7	5
18	236.6	155.2	3	3	3	5	7	7
19	385.4	295	3	3	3	7	3	3
20	261.1	184.9	3	3	3	7	3	5
21	203	126.6	3	3	3	7	3	7
22	481.2	376.8	3	3	3	7	5	3
23	312	228.8	3	3	3	7	5	5
24	233.2	152.6	3	3	3	7	5	7
25	624.6	500	3	3	3	7	7	3
26	388.1	294.2	3	3	3	7	7	5
27	277.6	191.3	3	3	3	7	7	7
28	176.6	100.9	3	3	5	3	3	3

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
29	148.2	63.6	3	3	5	3	3	5
30	129.1	28.3	3	3	5	3	3	7
31	319.2	224.2	3	3	5	3	5	3
32	245.5	147.4	3	3	5	3	5	5
33	194.9	84.7	3	3	5	3	5	7
34	532.1	408.6	3	3	5	3	7	3
35	390.5	273.2	3	3	5	3	7	5
36	291.6	169.8	3	3	5	3	7	7
37	307.3	215.1	3	3	5	5	3	3
38	237.6	142.1	3	3	5	5	3	5
39	188.6	82.1	3	3	5	5	3	7
40	449.4	338.1	3	3	5	5	5	3
41	334.6	225.8	3	3	5	5	5	5
42	254.5	138.4	3	3	5	5	5	7
43	561.8	522	3	3	5	5	7	3
44	479.3	351.3	3	3	5	5	7	5
45	351.5	233.2	3	3	5	5	7	7
46	502.5	386.1	3	3	5	7	3	3
47	371.3	259.7	3	3	5	7	3	5

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
48	278.4	162.2	3	3	5	7	3	7
49	644.1	508.6	3	3	5	7	5	3
50	467.9	342.9	3	3	5	7	5	5
51	344.1	218.3	3	3	5	7	5	7
52	855.6	691.8	3	3	5	7	7	3
53	612.1	468	3	3	5	7	7	5
54	441.2	302.8	3	3	5	7	7	7
55	188	104.3	3	3	7	3	3	3
56	162.7	58.8	3	3	7	3	3	5
57	139.8	20.7	3	3	7	3	3	7
58	352.3	246.2	3	3	7	3	5	3
59	291.8	170.2	3	3	7	3	5	5
60	236.6	104.3	3	3	7	3	5	7
61	596.7	458.4	3	3	7	3	7	3
62	483.3	337.1	3	3	7	3	7	5
63	379.4	230.4	3	3	7	3	7	7
64	337.6	235.9	3	3	7	5	3	3
65	280.3	163.1	3	3	7	5	3	5
66	226.7	100.1	3	3	7	5	3	7

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
67	501.5	377.5	3	3	7	5	5	3
68	409.2	274.2	3	3	7	5	5	5
69	324.3	183.6	3	3	7	5	5	7
70	745.5	589.1	3	3	7	5	7	3
71	600.6	440.7	3	3	7	5	7	5
72	467.8	309.3	3	3	7	5	7	7
73	561.8	432.7	3	3	7	7	3	3
74	457	319	3	3	7	7	3	5
75	359.4	218.6	3	3	7	7	3	7
76	724.9	573.7	3	3	7	7	5	3
77	585.3	429.7	3	3	7	7	5	5
78	456.8	301.8	3	3	7	7	5	7
79	968.1	784.6	3	3	7	7	7	3
80	776.3	595.6	3	3	7	7	7	5
81	600.5	427.1	3	3	7	7	7	7
82	252.3	181	3	5	3	3	3	3
83	272	192.4	3	5	3	3	3	5
84	291.1	194.7	3	5	3	3	3	7
85	348.4	261.8	3	5	3	3	5	3

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
86	323	237.4	3	5	3	3	5	5
87	312.5	220.8	3	5	3	3	5	7
88	492.5	384.7	3	5	3	3	7	3
89	399.7	304	3	5	3	3	7	5
90	357.9	259.8	3	5	3	3	7	7
91	340.7	257	3	5	3	5	3	3
92	319.3	234.1	3	5	3	5	3	5
93	308.9	219.3	3	5	3	5	3	7
94	436.7	338.5	3	5	3	5	5	3
95	370.2	278.7	3	5	3	5	5	5
96	340.1	245.4	3	5	3	5	5	7
97	580.4	461.5	3	5	3	5	7	3
98	446.6	345.1	3	5	3	5	7	5
99	385.4	284.2	3	5	3	5	7	7
100	473	370.1	3	5	3	7	3	3
101	389.9	295.9	3	5	3	7	3	5
102	350.5	255.7	3	5	3	7	3	7
103	568.7	452.8	3	5	3	7	5	3
104	440.7	340.3	3	5	3	7	5	5

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
105	381.4	281.7	3	5	3	7	5	7
106	712	575.7	3	5	3	7	7	3
107	516.9	406.4	3	5	3	7	7	5
108	426.5	320.4	3	5	3	7	7	7
109	223.2	141.5	3	5	5	3	3	3
110	235.8	140.8	3	5	5	3	3	5
111	243.5	130.4	3	5	5	3	3	7
112	365.4	264.8	3	5	5	3	5	3
113	333.1	224.6	3	5	5	3	5	5
114	311.4	186.8	3	5	5	3	5	7
115	578.5	499.2	3	5	5	3	7	3
116	478.6	350.4	3	5	5	3	7	5
117	410.1	271.9	3	5	5	3	7	7
118	353.8	255.7	3	5	5	5	3	3
119	325.4	219.3	3	5	5	5	3	5
120	304.1	184.2	3	5	5	5	3	7
121	495.7	378.7	3	5	5	5	5	3
122	422.4	301.9	3	5	5	5	5	5
123	371.5	240.5	3	5	5	5	5	7

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
124	708.2	562.7	3	5	5	5	7	3
125	567.5	428.4	3	5	5	5	7	5
126	469.9	325.3	3	5	5	5	7	7
127	549.1	426.7	3	5	5	7	3	3
128	459.3	336.8	3	5	5	7	3	5
129	394.6	264.3	3	5	5	7	3	7
130	690.5	549.2	3	5	5	7	5	3
131	555.9	420.1	3	5	5	7	5	5
132	461.4	320.4	3	5	5	7	5	7
133	902.1	732.5	3	5	5	7	7	3
134	700.3	545.2	3	5	5	7	7	5
135	559.6	404.9	3	5	5	7	7	7
136	214.2	128.2	3	5	7	3	3	3
137	220.7	111.1	3	5	7	3	3	5
138	224.7	97.8	3	5	7	3	3	7
139	379.3	270.1	3	5	7	3	5	3
140	351.2	222.4	3	5	7	3	5	5
141	325.8	181.5	3	5	7	3	5	7
142	624.7	482.3	3	5	7	3	7	3

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
143	544.2	389.4	3	5	7	3	7	5
144	471.5	307.6	3	5	7	3	7	7
145	364.4	259.8	3	5	7	5	3	3
146	339.2	215.4	3	5	7	5	3	5
147	313.7	177.3	3	5	7	5	3	7
148	528.6	401.3	3	5	7	5	5	3
149	468.9	326.5	3	5	7	5	5	5
150	413.9	260.8	3	5	7	5	5	7
151	773.4	613	3	5	7	5	7	3
152	661.5	493	3	5	7	5	7	5
153	559.7	386.5	3	5	7	5	7	7
154	588.8	456.5	3	5	7	7	3	3
155	516.4	371.3	3	5	7	7	3	5
156	447.3	295.9	3	5	7	7	3	7
157	752.2	597.6	3	5	7	7	5	3
158	645.2	482	3	5	7	7	5	5
159	546.6	379	3	5	7	7	5	7
160	996	808.4	3	5	7	7	7	3
161	837.1	647.9	3	5	7	7	7	5

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
162	692	504.3	3	5	7	7	7	7
163	393.1	292.6	3	7	3	3	3	3
164	464.7	359.8	3	7	3	3	3	5
165	502	388.1	3	7	3	3	3	7
166	479.4	375.2	3	7	3	3	5	3
167	515.7	404.5	3	7	3	3	5	5
168	533.2	414.2	3	7	3	3	5	7
169	623.4	498.5	3	7	3	3	7	3
170	592.3	471.3	3	7	3	3	7	5
171	579	453.2	3	7	3	3	7	7
172	471.5	369.5	3	7	3	5	3	3
173	511.9	401.3	3	7	3	5	3	5
174	529.9	412.7	3	7	3	5	3	7
175	567.6	452	3	7	3	5	5	3
176	562.9	445.9	3	7	3	5	5	5
177	561	438.7	3	7	3	5	5	7
178	711.3	575.3	3	7	3	5	7	3
179	639.3	512.4	3	7	3	5	7	5
180	606.7	477.6	3	7	3	5	7	7

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
181	603.8	484	3	7	3	7	3	3
182	582.6	463.3	3	7	3	7	3	5
183	571.6	449.2	3	7	3	7	3	7
184	699.6	566.5	3	7	3	7	5	3
185	633.4	507.6	3	7	3	7	5	5
186	602.4	475.1	3	7	3	7	5	7
187	842.8	689.6	3	7	3	7	7	3
188	709.5	573.9	3	7	3	7	7	5
189	647.9	513.8	3	7	3	7	7	7
190	293.1	202.3	3	7	5	3	3	3
191	367.6	256.1	3	7	5	3	3	5
192	417.1	283.3	3	7	5	3	3	7
193	435	325.5	3	7	5	3	5	3
194	464.6	340	3	7	5	3	5	5
195	484.7	339.8	3	7	5	3	5	7
196	647.9	509.9	3	7	5	3	7	3
197	610.1	465.8	3	7	5	3	7	5
198	584.5	424.8	3	7	5	3	7	7
199	423.6	316.5	3	7	5	5	3	3

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
200	457.2	334.7	3	7	5	5	3	5
201	477.9	337.2	3	7	5	5	3	7
202	565.3	439.5	3	7	5	5	5	3
203	554	418.3	3	7	5	5	5	5
204	545.2	393.4	3	7	5	5	5	7
205	777.6	623.4	3	7	5	5	7	3
206	699.1	543.8	3	7	5	5	7	5
207	644.6	478.2	3	7	5	5	7	7
208	618.9	487.5	3	7	5	7	3	3
209	591.2	452.2	3	7	5	7	3	5
210	568.8	417.3	3	7	5	7	3	7
211	760.1	610	3	7	5	7	5	3
212	687.5	535.5	3	7	5	7	5	5
213	635.6	473.3	3	7	5	7	5	7
214	971.6	793.2	3	7	5	7	7	3
215	832	660.6	3	7	5	7	7	5
216	734.5	557.8	3	7	5	7	7	7
217	254.9	163.9	3	7	7	3	3	3
218	309.4	189.3	3	7	7	3	3	5

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
219	354.9	213.8	3	7	7	3	3	7
220	419.3	305.9	3	7	7	3	5	3
221	439.4	300.7	3	7	7	3	5	5
222	456.1	291.4	3	7	7	3	5	7
223	665.2	518.1	3	7	7	3	7	3
224	633.2	467.7	3	7	7	3	7	5
225	604.1	423.5	3	7	7	3	7	7
226	405.2	295.5	3	7	7	5	3	3
227	428.2	293.7	3	7	7	5	3	5
228	444.7	293.2	3	7	7	5	3	7
229	568.9	437.1	3	7	7	5	5	3
230	557.5	404.8	3	7	7	5	5	5
231	545.1	376.7	3	7	7	5	5	7
232	814	648.8	3	7	7	5	7	3
233	750.7	571.3	3	7	7	5	7	5
234	692.5	502.4	3	7	7	5	7	7
235	629.7	492.3	3	7	7	7	3	3
236	605.6	449.6	3	7	7	7	3	5
237	579.1	411.8	3	7	7	7	3	7

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
238	792.7	633.3	3	7	7	7	5	3
239	734.1	560.2	3	7	7	7	5	5
240	678.4	494.9	3	7	7	7	5	7
241	1036.7	844.2	3	7	7	7	7	3
242	926.4	726.1	3	7	7	7	7	5
243	825.1	620.2	3	7	7	7	7	7
244	257.4	187.2	5	3	3	3	3	3
245	280	198.7	5	3	3	3	3	5
246	290.3	202.1	5	3	3	3	3	7
247	353.7	267.6	5	3	3	3	5	3
248	331.2	243.3	5	3	3	3	5	5
249	320.8	228.2	5	3	3	3	5	7
250	498	390.3	5	3	3	3	7	3
251	407.9	309.4	5	3	3	3	7	5
252	365.6	267.2	5	3	3	3	7	7
253	345.9	266.5	5	3	3	5	3	3
254	327.3	240.5	5	3	3	5	3	5
255	318.4	226.7	5	3	3	5	3	7
256	442	346.8	5	3	3	5	5	3

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
257	378.4	284.9	5	3	3	5	5	5
258	348.7	252.8	5	3	3	5	5	7
259	586	469	5	3	3	5	7	3
260	454.9	350.8	5	3	3	5	7	5
261	393.5	291.6	5	3	3	5	7	7
262	478.3	380.2	5	3	3	7	3	3
263	398	302.8	5	3	3	7	3	5
264	360.1	263.2	5	3	3	7	3	7
265	574.1	461.2	5	3	3	7	5	3
266	448.9	346.9	5	3	3	7	5	5
267	390.2	289.1	5	3	3	7	5	7
268	717.5	583.5	5	3	3	7	7	3
269	525.1	412.5	5	3	3	7	7	5
270	434.8	327.8	5	3	3	7	7	7
271	225.9	143.8	5	3	5	3	3	3
272	241.4	145.1	5	3	5	3	3	5
273	250.7	136.4	5	3	5	3	3	7
274	368.6	267.1	5	3	5	3	5	3
275	338.8	229	5	3	5	3	5	5

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
276	316.9	192.9	5	3	5	3	5	7
277	581.5	451.5	5	3	5	3	7	3
278	484	354.8	5	3	5	3	7	5
279	414.6	277.9	5	3	5	3	7	7
280	356.6	258.1	5	3	5	5	3	3
281	331.1	223.7	5	3	5	5	3	5
282	311.7	190.2	5	3	5	5	3	7
283	498.8	381	5	3	5	5	5	3
284	428.1	307.3	5	3	5	5	5	5
285	377.6	246.5	5	3	5	5	5	7
286	711.2	565	5	3	5	5	7	3
287	572.9	432.8	5	3	5	5	7	5
288	475.1	331.3	5	3	5	5	7	7
289	552	429	5	3	5	7	3	3
290	465	341.2	5	3	5	7	3	5
291	402.4	270.3	5	3	5	7	3	7
292	693.5	551.5	5	3	5	7	5	3
293	561.6	424.5	5	3	5	7	5	5
294	468	326.4	5	3	5	7	5	7

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
295	905.1	734.8	5	3	5	7	7	3
296	705.8	549.6	5	3	5	7	7	5
297	565.3	410.9	5	3	5	7	7	7
298	216.6	129.5	5	3	7	3	3	3
299	225.1	113.9	5	3	7	3	3	5
300	230.1	102.2	5	3	7	3	3	7
301	381.1	271.4	5	3	7	3	5	3
302	354.5	225.3	5	3	7	3	5	5
303	328.2	186	5	3	7	3	5	7
304	625.6	483.6	5	3	7	3	7	3
305	546.4	392.2	5	3	7	3	7	5
306	472.2	312.1	5	3	7	3	7	7
307	336.5	261.1	5	3	7	5	3	3
308	343.5	218.2	5	3	7	5	3	5
309	319.7	181.7	5	3	7	5	3	7
310	530.4	402.6	5	3	7	5	5	3
311	472.4	329.3	5	3	7	5	5	5
312	417.4	265.2	5	3	7	5	5	7
313	774.5	614.3	5	3	7	5	7	3

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
314	664	495.9	5	3	7	5	7	5
315	561.4	391	5	3	7	5	7	7
316	590.9	457.8	5	3	7	7	3	3
317	520.7	374.1	5	3	7	7	3	5
318	453.6	300.3	5	3	7	7	3	7
319	754	598.9	5	3	7	7	5	3
320	648.9	484.8	5	3	7	7	5	5
321	500.8	383.5	5	3	7	7	5	7
322	997.3	809.7	5	3	7	7	7	3
323	839.9	650.7	5	3	7	7	7	5
324	694.7	508.8	5	3	7	7	7	7
325	344.8	261.3	5	5	3	3	3	3
326	408.6	310.3	5	5	3	3	3	5
327	437.8	331	5	5	3	3	3	7
328	441	342.7	5	5	3	3	5	3
329	459.7	335.1	5	5	3	3	5	5
330	468.7	357.1	5	5	3	3	5	7
331	585.2	465.7	5	5	3	3	7	3
332	536.3	421.7	5	5	3	3	7	5

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
333	514.1	396.1	5	5	3	3	7	7
334	433.3	340.6	5	5	3	5	3	3
335	455.9	352	5	5	3	5	3	5
336	465.9	355.7	5	5	3	5	3	7
337	429.3	421.7	5	5	3	5	5	3
338	506.9	396.6	5	5	3	5	5	5
339	496.7	381.7	5	5	3	5	5	7
340	673.1	544.2	5	5	3	5	7	3
341	583.3	463	5	5	3	5	7	5
342	541.9	420.6	5	5	3	5	7	7
343	565.6	455	5	5	3	7	3	3
344	526.6	414.3	5	5	3	7	3	5
345	507.6	392.2	5	5	3	7	3	7
346	661.3	536.5	5	5	3	7	5	3
347	577.4	458.6	5	5	3	7	5	5
348	538.2	418.1	5	5	3	7	5	7
349	804.6	658.9	5	5	3	7	7	3
350	653.5	524.6	5	5	3	7	7	5
351	583.3	456.8	5	5	3	7	7	7

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
352	272.4	184.4	5	5	5	3	3	3
353	329.1	222.1	5	5	5	3	3	5
354	336.5	238.3	5	5	5	3	3	7
355	414.7	307.6	5	5	5	3	5	3
356	426.4	305.9	5	5	5	3	5	5
357	433.9	294.7	5	5	5	3	5	7
358	627.8	492	5	5	5	3	7	3
359	572	431.8	5	5	5	3	7	5
360	532.7	379.7	5	5	5	3	7	7
361	403.1	298.6	5	5	5	5	3	3
362	418.8	300.7	5	5	5	5	3	5
363	427.6	292.1	5	5	5	5	3	7
364	545	421.6	5	5	5	5	5	3
365	515.8	384.3	5	5	5	5	5	5
366	494.6	348.4	5	5	5	5	5	7
367	757.5	605.5	5	5	5	5	7	3
368	660.9	509.8	5	5	5	5	7	5
369	593.1	433.2	5	5	5	5	7	7
370	598.4	469.6	5	5	5	7	3	3

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
371	552.8	418.2	5	5	5	7	3	5
372	518.6	372.2	5	5	5	7	3	7
373	739.8	592.1	5	5	5	7	5	3
374	649.4	501.5	5	5	5	7	5	5
375	585.1	428.3	5	5	5	7	5	7
376	951.5	775.3	5	5	5	7	7	3
377	793.8	626.5	5	5	5	7	7	5
378	683.2	512.7	5	5	5	7	7	7
379	243	153.3	5	5	7	3	3	3
380	283.6	166.1	5	5	7	3	3	5
381	317	179.3	5	5	7	3	3	7
382	408	295.2	5	5	7	3	5	3
383	414	277.4	5	5	7	3	5	5
384	417.6	263	5	5	7	3	5	7
385	653.6	507.4	5	5	7	3	7	3
386	607.2	444.4	5	5	7	3	7	5
387	563.8	389	5	5	7	3	7	7
388	393.3	284.9	5	5	7	5	3	3
389	402.5	270.4	5	5	7	5	3	5

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
390	407.1	258.8	5	5	7	5	3	7
391	557.5	426.4	5	5	7	5	5	3
392	532.1	381.5	5	5	7	5	5	5
393	506.9	342.2	5	5	7	5	5	7
394	802.4	638.1	5	5	7	5	7	3
395	724.7	548	5	5	7	5	7	5
396	652.7	468	5	5	7	5	7	7
397	617.9	481.6	5	5	7	7	3	3
398	579.9	426.3	5	5	7	7	3	5
399	541.5	377.4	5	5	7	7	3	7
400	781.2	622.7	5	5	7	7	5	3
401	708.7	537	5	5	7	7	5	5
402	640.3	460.5	5	5	7	7	5	7
403	1025.1	833.6	5	5	7	7	7	3
404	900.5	702.9	5	5	7	7	7	5
405	785.6	585.8	5	5	7	7	7	7
406	475.3	373.1	5	7	3	3	3	3
407	600.8	477.7	5	7	3	3	3	5
408	658.5	524.1	5	7	3	3	3	7

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
409	571.6	455.7	5	7	3	3	5	3
410	651.9	522.4	5	7	3	3	5	5
411	689.4	550.1	5	7	3	3	5	7
412	715.7	579	5	7	3	3	7	3
413	728.5	589.1	5	7	3	3	7	5
414	735.1	589.1	5	7	3	3	7	7
415	563.7	452.1	5	7	3	5	3	3
416	648.1	519.4	5	7	3	5	3	5
417	686.6	548.7	5	7	3	5	3	7
418	659.9	534.3	5	7	3	5	5	3
419	699.1	563.8	5	7	3	5	5	5
420	717.4	574.7	5	7	3	5	5	7
421	803.6	657.2	5	7	3	5	7	3
422	775.5	630.3	5	7	3	5	7	5
423	762.9	613.6	5	7	3	5	7	7
424	696.1	597.2	5	7	3	7	3	3
425	718.8	581.5	5	7	3	7	3	5
426	728.3	585.2	5	7	3	7	3	7
427	791.9	649.3	5	7	3	7	5	3

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
428	769.6	625.8	5	7	3	7	5	5
429	759	611.1	5	7	3	7	5	7
430	935.1	772	5	7	3	7	7	3
431	845.7	691.9	5	7	3	7	7	5
432	804.3	649.8	5	7	3	7	7	7
433	342.1	245	5	7	5	3	3	3
434	460.7	337.2	5	7	5	3	3	5
435	560.3	390.9	5	7	5	3	3	7
436	484.1	368.2	5	7	5	3	5	3
437	557.7	421.1	5	7	5	3	5	5
438	607.6	447.3	5	7	5	3	5	7
439	697.1	552.6	5	7	5	3	7	3
440	703.2	546.9	5	7	5	3	7	5
441	707.1	532.3	5	7	5	3	7	7
442	472.7	359.2	5	7	5	5	3	3
443	550.4	415.8	5	7	5	5	3	5
444	601.4	444.7	5	7	5	5	3	7
445	614.4	482.2	5	7	5	5	5	3
446	647.2	488.4	5	7	5	5	5	5

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
447	668.4	500.9	5	7	5	5	5	7
448	826.8	666.1	5	7	5	5	7	3
449	729.2	624.9	5	7	5	5	7	5
450	767.6	585.7	5	7	5	5	7	7
451	668	530.2	5	7	5	7	3	3
452	684.4	533.3	5	7	5	7	3	5
453	692.5	524.8	5	7	5	7	3	7
454	809.2	652.7	5	7	5	7	5	3
455	780.8	616.6	5	7	5	7	5	5
456	759.1	580.8	5	7	5	7	5	7
457	1020.8	835.9	5	7	5	7	7	3
458	925.2	741.7	5	7	5	7	7	5
459	857.7	665.3	5	7	5	7	7	7
460	283.6	189	5	7	7	3	3	3
461	372.4	244.1	5	7	7	3	3	5
462	447.8	294.9	5	7	7	3	3	7
463	448	330.9	5	7	7	3	5	3
464	502.2	355.5	5	7	7	3	5	5
465	548.5	378.6	5	7	7	3	5	7

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
466	693.9	543.1	5	7	7	3	7	3
467	696.1	522.5	5	7	7	3	7	5
468	696.4	504.7	5	7	7	3	7	7
469	434	320.5	5	7	7	5	3	3
470	491.4	348.5	5	7	7	5	3	5
471	538.3	374.4	5	7	7	5	3	7
472	597.8	462.1	5	7	7	5	5	3
473	620.6	459.6	5	7	7	5	5	5
474	638.1	457.9	5	7	7	5	5	7
475	842.9	673.8	5	7	7	5	7	3
476	813.7	826.1	5	7	7	5	7	5
477	785.4	583.6	5	7	7	5	7	7
478	658.7	517.3	5	7	7	7	3	3
479	669	504.4	5	7	7	7	3	5
480	673	493	5	7	7	7	3	7
481	821.7	658.3	5	7	7	7	5	3
482	797.5	315	5	7	7	7	5	5
483	772	576.1	5	7	7	7	5	7
484	1065.7	869.2	5	7	7	7	7	3

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
485	989.7	781	5	7	7	7	7	5
486	918.5	701.4	5	7	7	7	7	7
487	396.4	304.7	7	3	3	3	3	3
488	485	375.7	7	3	3	3	3	5
489	525.5	406.5	7	3	3	3	3	7
490	492.8	386.5	7	3	3	3	5	3
491	536.3	420.2	7	3	3	3	5	5
492	555.8	432.6	7	3	3	3	5	7
493	637.1	510	7	3	3	3	7	3
494	612.9	486.5	7	3	3	3	7	5
495	600.8	471.6	7	3	3	3	7	7
496	484.9	386.4	7	3	3	5	3	3
497	532.3	417.4	7	3	3	5	3	5
498	553.7	431.2	7	3	3	5	3	7
499	581.1	467.5	7	3	3	5	5	3
500	583.4	461.7	7	3	3	5	5	5
501	583.9	457.2	7	3	3	5	5	7
502	725.1	589.9	7	3	3	5	7	3
503	659.9	527.8	7	3	3	5	7	5

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
504	628.7	496.1	7	3	3	5	7	7
505	617.4	502.7	7	3	3	7	3	3
506	603.1	479.7	7	3	3	7	3	5
507	595.6	467.7	7	3	3	7	3	7
508	713.2	583.8	7	3	3	7	5	3
509	654	523.8	7	3	3	7	5	5
510	625.6	493.6	7	3	3	7	5	7
511	856.6	705.9	7	3	3	7	7	3
512	730.1	589.6	7	3	3	7	7	5
513	670.2	532.3	7	3	3	7	7	7
514	300.1	208.1	7	3	5	3	3	3
515	381.6	267.2	7	3	5	3	3	5
516	435.6	298.3	7	3	5	3	3	7
517	442.7	331.4	7	3	5	3	5	3
518	478.9	351.1	7	3	5	3	5	5
519	501.7	354.7	7	3	5	3	5	7
520	655.6	515.8	7	3	5	3	7	3
521	624.2	476.9	7	3	5	3	7	5
522	599.8	439.8	7	3	5	3	7	7

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
523	430.7	322.3	7	3	5	5	3	3
524	472.3	345.8	7	3	5	5	3	5
525	496.8	352.1	7	3	5	5	3	7
526	572.8	445.3	7	3	5	5	5	3
527	568.3	429.4	7	3	5	5	5	5
528	562.7	408.4	7	3	5	5	5	7
529	785.3	629.2	7	3	5	5	7	3
530	713.1	554.9	7	3	5	5	7	5
531	660.5	493.2	7	3	5	5	7	7
532	626	493.3	7	3	5	7	3	3
533	605.3	463.3	7	3	5	7	3	5
534	588.1	432.2	7	3	5	7	3	7
535	767.6	615.8	7	3	5	7	5	3
536	701.9	546.6	7	3	5	7	5	5
537	653.6	488.3	7	3	5	7	5	7
538	979.2	799	7	3	5	7	7	3
539	846.1	671.6	7	3	5	7	7	5
540	751	572.8	7	3	5	7	7	7
541	260	167.3	7	3	7	3	3	3

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
542	319.7	196.7	7	3	7	3	3	5
543	369.1	224.9	7	3	7	3	3	7
544	424.6	309.2	7	3	7	3	5	3
545	449.1	308.1	7	3	7	3	5	5
546	467.3	308.7	7	3	7	3	5	7
547	669.2	521.5	7	3	7	3	7	3
548	641.3	475.1	7	3	7	3	7	5
549	612.2	434.7	7	3	7	3	7	7
550	409.9	298.9	7	3	7	5	3	3
551	438.4	301.1	7	3	7	5	3	5
552	459.6	304.4	7	3	7	5	3	7
553	573.9	440.5	7	3	7	5	5	3
554	567.3	412.1	7	3	7	5	5	5
555	557.3	387.9	7	3	7	5	5	7
556	818.1	652.1	7	3	7	5	7	3
557	759.1	578.7	7	3	7	5	7	5
558	701.8	513.7	7	3	7	5	7	7
559	634.5	495.7	7	3	7	7	3	3
560	615.9	457	7	3	7	7	3	5

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
561	594.4	423	7	3	7	7	3	7
562	797.6	616.7	7	3	7	7	5	3
563	744.1	567.6	7	3	7	7	5	5
564	691.5	506.2	7	3	7	7	5	7
565	1040.9	847.6	7	3	7	7	7	3
566	935.2	733.5	7	3	7	7	7	5
567	835.6	631.5	7	3	7	7	7	7
568	483.5	379.6	7	5	3	3	3	3
569	613.1	487.3	7	5	3	3	3	5
570	672.7	535.1	7	5	3	3	3	7
571	579.7	461.8	7	5	3	3	5	3
572	664.3	531.9	7	5	3	3	5	5
573	703.4	561.1	7	5	3	3	5	7
574	723.9	585.3	7	5	3	3	7	3
575	740.9	598.5	7	5	3	3	7	5
576	748.7	600	7	5	3	3	7	7
577	571.2	460.7	7	5	3	5	3	3
578	660.4	528.9	7	5	3	5	3	5
579	700.8	559.8	7	5	3	5	3	7

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
580	668	542.3	7	5	3	5	5	3
581	711.4	573.4	7	5	3	5	5	5
582	731.4	585.8	7	5	3	5	5	7
583	811.9	664.9	7	5	3	5	7	3
584	787.9	639.7	7	5	3	5	7	5
585	776.6	624.6	7	5	3	5	7	7
586	704.3	577.2	7	5	3	7	3	3
587	731.2	591.2	7	5	3	7	3	5
588	742.7	596.3	7	5	3	7	3	7
589	800.1	658.6	7	5	3	7	5	3
590	782	635.4	7	5	3	7	5	5
591	773.1	622.2	7	5	3	7	5	7
592	943.4	780.9	7	5	3	7	7	3
593	858.1	701.4	7	5	3	7	7	5
594	818.1	600.9	7	5	3	7	7	7
595	346.4	248.5	7	5	5	3	3	3
596	469.1	343.9	7	5	5	3	3	5
597	551.5	399.8	7	5	5	3	3	7
598	488.7	371.8	7	5	5	3	5	3

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
599	566.4	427.8	7	5	5	3	5	5
600	618.5	456.2	7	5	5	3	5	7
601	701.8	556.2	7	5	5	3	7	3
602	711.9	553.6	7	5	5	3	7	5
603	717.3	541.3	7	5	5	3	7	7
604	477	362.7	7	5	5	5	3	3
605	558.8	422.5	7	5	5	5	3	5
606	612.8	453.6	7	5	5	5	3	7
607	618.9	485.7	7	5	5	5	5	3
608	655.8	506.1	7	5	5	5	5	5
609	679.4	509.9	7	5	5	5	5	7
610	831.4	669.6	7	5	5	5	7	3
611	800.8	631.6	7	5	5	5	7	5
612	777.9	594.7	7	5	5	5	7	7
613	672.3	533.7	7	5	5	7	3	3
614	692.9	540	7	5	5	7	3	5
615	704.1	533.7	7	5	5	7	3	7
616	813.7	656.2	7	5	5	7	5	3
617	789.4	623.3	7	5	5	7	5	5

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
618	770.3	589.8	7	5	5	7	5	7
619	1025.4	839.4	7	5	5	7	7	3
620	933.9	748.3	7	5	5	7	7	5
621	868.3	674.3	7	5	5	7	7	7
622	286.4	191	7	5	7	3	3	3
623	378.4	248.7	7	5	7	3	3	5
624	456.6	301.7	7	5	7	3	3	7
625	451.4	333	7	5	7	3	5	3
626	508.5	360.1	7	5	7	3	5	5
627	556.6	385.4	7	5	7	3	5	7
628	697	545.2	7	5	7	3	7	3
629	701.8	527	7	5	7	3	7	5
630	703	511.5	7	5	7	3	7	7
631	436.7	322.6	7	5	7	5	3	3
632	498.3	353	7	5	7	5	3	5
633	547.2	381.2	7	5	7	5	3	7
634	600.9	464.2	7	5	7	5	5	3
635	626.8	464.1	7	5	7	5	5	5
636	646.4	464.6	7	5	7	5	5	7

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
637	845.8	675.9	7	5	7	5	7	3
638	819.4	630.6	7	5	7	5	7	5
639	792.4	590.4	7	5	7	5	7	7
640	661.4	519.4	7	5	7	7	3	3
641	675	508.9	7	5	7	7	3	5
642	682.1	499.8	7	5	7	7	3	7
643	824.7	660.4	7	5	7	7	5	3
644	803.6	619.6	7	5	7	7	5	5
645	780.6	582.9	7	5	7	7	5	7
646	1068.6	871.3	7	5	7	7	7	3
647	995.5	785.5	7	5	7	7	7	5
648	925.9	708.2	7	5	7	7	7	7
649	613.5	491.8	7	7	3	3	3	3
650	804.7	654.3	7	7	3	3	3	5
651	892.8	727.5	7	7	3	3	3	7
652	709.9	574.7	7	7	3	3	5	3
653	855.8	698.9	7	7	3	3	5	5
654	923.5	753.5	7	7	3	3	5	7
655	854	698.4	7	7	3	3	7	3

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
656	932.4	765.5	7	7	3	3	7	5
657	969.1	792.5	7	7	3	3	7	7
658	701.9	572.2	7	7	3	5	3	3
659	852	695.9	7	7	3	5	3	5
660	920.9	752.2	7	7	3	5	3	7
661	798.1	654.5	7	7	3	5	5	3
662	903	740.3	7	7	3	5	5	5
663	951.5	778.1	7	7	3	5	5	7
664	941.9	777.5	7	7	3	5	7	3
665	979.4	806.7	7	7	3	5	7	5
666	996.9	817	7	7	3	5	7	7
667	834.3	688.8	7	7	3	7	3	3
668	922.7	758.1	7	7	3	7	3	5
669	962.8	788.7	7	7	3	7	3	7
670	930.1	770.8	7	7	3	7	5	3
671	973.5	802.3	7	7	3	7	5	5
672	993.2	814.6	7	7	3	7	5	7
673	1073.4	893.4	7	7	3	7	7	3
674	1049.7	868.4	7	7	3	7	7	5

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
675	1038.5	853.3	7	7	3	7	7	7
676	415.8	308.9	7	7	5	3	3	3
677	600.3	458.6	7	7	5	3	3	5
678	725	551.8	7	7	5	3	3	7
679	557.8	432.1	7	7	5	3	5	3
680	697.3	542.5	7	7	5	3	5	5
681	791.9	608.2	7	7	5	3	5	7
682	770.8	616.5	7	7	5	3	7	3
683	842.8	668.3	7	7	5	3	7	5
684	891.3	693.3	7	7	5	3	7	7
685	546.3	423.1	7	7	5	5	3	3
686	689.9	537.2	7	7	5	5	3	5
687	786.2	605.6	7	7	5	5	3	7
688	688.1	546.1	7	7	5	5	5	3
689	786.7	620.8	7	7	5	5	5	5
690	852.9	661.9	7	7	5	5	5	7
691	900.5	730	7	7	5	5	7	3
692	931.8	746.3	7	7	5	5	7	5
693	951.9	746.7	7	7	5	5	7	7

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
694	741.7	594.1	7	7	5	7	3	3
695	824	654.7	7	7	5	7	3	5
696	877.6	685.8	7	7	5	7	3	7
697	882.9	716.6	7	7	5	7	5	3
698	920.4	738	7	7	5	7	5	5
699	943.8	741.8	7	7	5	7	5	7
700	1094.5	899.8	7	7	5	7	7	3
701	1064.8	863.1	7	7	5	7	7	5
702	1042.4	826.3	7	7	5	7	7	7
703	326.9	226.6	7	7	7	3	3	3
704	467	326.5	7	7	7	3	3	5
705	587.6	416.9	7	7	7	3	3	7
706	491.3	368.5	7	7	7	3	5	3
707	596.7	437.9	7	7	7	3	5	5
708	687.7	500.6	7	7	7	3	5	7
709	737.2	580.7	7	7	7	3	7	3
710	790.5	604.8	7	7	7	3	7	5
711	835.4	626.7	7	7	7	3	7	7
712	477.3	358.2	7	7	7	5	3	3

Continued on next page

Table A.2 – continued from previous page

Run #	Y_{20} (m)	Y_{10} (m)	X_1 (deg)	X_2 (deg)	X_3 (nm)	X_4 (deg)	X_5 (deg)	X_6 (nm)
713	586	430.8	7	7	7	5	3	5
714	678.3	496.4	7	7	7	5	3	7
715	641	499.7	7	7	7	5	5	3
716	715.1	541.9	7	7	7	5	5	5
717	777.7	579.9	7	7	7	5	5	7
718	886.2	711.4	7	7	7	5	7	3
719	908.2	708.4	7	7	7	5	7	5
720	924.8	705.6	7	7	7	5	7	7
721	702.1	554.9	7	7	7	7	3	3
722	763.8	586.7	7	7	7	7	3	5
723	813.4	615	7	7	7	7	3	7
724	865	696	7	7	7	7	5	3
725	892.2	697.4	7	7	7	7	5	5
726	912	698.1	7	7	7	7	5	7
727	1109.1	906.8	7	7	7	7	7	3
728	1184.3	863.3	7	7	7	7	7	5
729	1058.3	823.4	7	7	7	7	7	7

Bibliography

[1] Faa creates commercial drone pilot's license (part 107): American drone operators will be able to apply for a drone pilot's license starting in august 2016. Marketwired, June 2016.

[2] M. Lisa and J. Staub. Ea-18g sea trials, flying qualities, performance and precision approach and landing systems. In *2008 Society of Experimental Test Pilots Annual Symposium*, 2008.

[3] Airborne software assurance. advisory circular no: 20-115c. U.S. Department of Transportation, Federal Aviaion Administration, 21 June, 2013.

[4] Teal group releases study, 'world unmanned aerial vehicle systems, market profile and forecast 2015.'. Manufacturing Close - Up, August 19, 2015.

[5] Matt Webster, Neil Cameron, Michael Fisher, and Mike Jump. Generating certification evidence for autonomous unmanned aircraft using model checking and simulation. *Journal of Aerospace Information Systems*, 11(5):258–279, 2014.

[6] R. E. Weibel and R. J. Hansman. Safety considerations for operation on unmanned aerial vehicles in the national airspace system. *Massachusetts Inst. of Technology International Center for Air Transportation*, TR-ICAT 2005-1, Cambridge, MA. 2005.

[7] Bcar section a: Airworthiness procedures where the caa has primary responsibility for type approval of a product. Civil Aviation Authority, London, Oct, 2011. CAP 553.

[8] Matthew Clark, Jim Alley, Paul J. Deal, Jeffrey C. Depriest, Eric Hansen, Connie Heitmeyer, Richard Nameth, Marc Steinberg, Craig Turner, Stuart Young, Darryl Ahner, Kelly Alonzo, Barry A. Bodt, P. F. Friesen, Jim Horris, Jonathan A. Hoffman, Kerianne H. Gross, Laura Humphrey, Marshal Childers, and Michael Corey. Autonomy community of interest (coi) test and evaluation, verification and validation (tevv) working group: Technology investment strategy 2015-2018. Technical report, Office of the Assistant Secretary of Defense for Research and Engineering, 2015.

[9] Navair airworthiness and cybersafe process manual, navair manual m-13034.1. NAVAIR, 2016.

[10] Naval air training and operations procedures standardization (natops) general flight and operating instructions. Commander Naval Air Forces, 2016.

[11] Operation of the defense acquisition system. Department of Defense Instruction Number 5000.02T, 2020.

[12] Autonomy in weapon systems. Department of Defense Directive Number 3000.9, 2012. Incorporating Change 1, May 8, 2017.

[13] Assistant secretary of defense for research and engineering guide, technology readiness assessment (tra). Asistant Secretry of Defencse for Research and Engineering (ASD(R&E)), 2011.

[14] John C. Mankins. Technology readiness levels. NASA Office of Space Access and Technology White Paper, 1995.

[15] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, Kenny Lau, Celia Oakley, Mark Palatucci, Vaughan Pratt, Pascal Stang, Sven Strohband, Cedric Dupont, Lars-Erik Jendrossek, Christian Koelen, Charles Markey, Carlo Rummel, Joe van Niekerk, Eric Jensen, Philippe Alessandrini, Gary Bradski, Bob Davies, Scott Ettinger, Adrian Kaehler, Ara Nefian, and Pamela Mahoney. Stanley: The robot that won the darpa grand challenge. *Journal of field robotics*, 23(9):661–692, 2006.

[16] Investigation - pe 16-007, odi resume. National Highway Traffic Safety Administration, US Department of Transportation, 2016.

[17] Brian A. Browne. Self-driving cars: On the road to a new regulatory era. *Journal of law, technology & the Internet*, 8:1, 2017.

[18] Matthew L. Roth. Regulating the future: Autonomous vehicles and the role of government. *Iowa law review*, 105(3):1411–1446, 2020.

[19] Kerianne H. Gross, Aaron W. Fifarek, and Jonathan A. Hoffman. Incremental formal methods based design approach demonstrated on a coupled tanks control system. In *2016 IEEE 17th International Symposium on High Assurance Systems Engineering (HASE)*, pages 181–188. IEEE, 2016.

[20] Jay Abraham. Verification and validation spanning models to code. In *AIAA Modeling and Simulation Technologies Conference*, 2015.

[21] Stephen Blanchette. Giant slayer: Will you let software be david to your goliath system? *Journal of Aerospace Information Systems*, 13(10):407–417, 2016.

[22] M. Muli, V. Moudgal, and J. Allen. Best practices and recommendations for model-based development process. *SAE Technical Paper 2015-01-2529*, 2015.

[23] Brett W. Israelsen, Nisar R. Ahmed, Kenneth Center, Roderick Green, and Winston Bennett. Towards adaptive training of agent-based sparring partners for fighter pilots. In *AIAA Information Systems-AIAA Infotech @ Aerospace*, 2017.

[24] Michael Fisher, Louise Dennis, and Matt Webster. Verifying autonomous systems. *Communications of the Association for Computing Machinery*, 56(9), 2013-09-01.

[25] Eric Tobias, Mark Tischler, Tom Berger, and Steven G. Hagerott. Full flight-envelope simulation and piloted fidelity assessment of a business jet using a model stitching architecture. In *AIAA Modeling and Simulation Technologies Conference*, 2015.

[26] Ulrich Eisemann and Jace L Allen. New requirement-definition and verification techniques according to do-178c, do-331, and do-333. In *AIAA Infotech@ Aerospace*, 2016.

[27] Tom Berger, Mark Tischler, Steven G. Hagerott, M Christopher Cotting, William R. Gray, James Gresham, Justin George, Kyle Krogh, Alessandro D'Argenio, and Justin Howland. Development and validation of a flight-identified full-envelope business jet simulation model using a stitching architecture. In *AIAA Modeling and Simulation Technologies Conference*, 2017.

[28] Shawn M. Walker, Jinjun Shan, and Lei Liu. Dimmacss-stage: a distributed intelligence model for a multi-agent control system using simulink and the stage robotic simulator. In *AIAA Modeling and Simulation Technologies Conference*, 2014.

[29] Swaroop A. Hangal, Bharat Tak, and Hemendra Arya. Distributed hardware-in-loop simulations for multiple autonomous aerial vehicles. In *AIAA Modeling and Simulation Technologies Conference*, 2015.

[30] Alaa El-Dien Shawky Mohamedy, Amgad M. Aly, and Amr H. Elnashar. Modeling and simulation hardware-in-the-loop for unmanned aerial vehicle. In *AIAA Modeling and Simulation Technologies Conference*, 2016.

[31] C. Baier and J Katoen. Principles of model checking. The MIT Press. Cambridge, Massachusetts, 2008.

[32] A. Kane. Runtime monitoring for safety-critical embedded systems (doctoral dissertation). electrical and computer engineering. *Carnegie Mellon University. Pittsburgh, PA*, 2015.

[33] Matthew Coombes, Wen-Hua Chen, and Peter Render. Site selection during unmanned aerial system forced landings using decision-making bayesian networks. *Journal of Aerospace Information Systems*, 13(12):491–495, 2016.

[34] Kerianne H. Gross, Matthew Clark, Jonathan A. Hoffman, Aaron Fifarek, Kuldip Rattan, Eric Swenson, Michael Whalen, and Lucas Wagner. Formally verified run time assurance architecture of a 6u cubesat attitude control system. In *AIAA Infotech @ Aerospace*, 2016.

[35] Christoph Torens, Florian Adolf, Peter Faymonville, and Sebastian Schirmer. Towards intelligent system health management using runtime monitoring. In *AIAA Information Systems-AIAA Infotech @ Aerospace*, 2017.

[36] Kerianne H. Gross, Matthew A. Clark, Jonathan A. Hoffman, Eric D. Swenson, and Aaron W. Fifarek. Run-time assurance and formal methods analysis nonlinear system applied to nonlinear system control. *Journal of Aerospace Information Systems*, 14(4):232–246, 2017.

[37] John Schierman, David Ward, Brian Dutoi, Anthony Aiello, John Berryman, Michael DeVore, Walter Storm, and Jason Wadley. Run-time verification and validation for safety-critical flight control systems. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008.

[38] Matthew Lichter, Alec Bateman, and Gary Balas. Flight test evaluation of a run-time stability margin estimation tool. In *AIAA Guidance, Navigation, and Control Conference*, 2009.

[39] Gregg Rabideau, Steve Chien, and David Mclaren. Onboard run-time goal selection for autonomous operations. In *SpaceOps 2010 Conference*, 2010.

[40] Belal Sababha, Hong Chul Yang, and Osamah Rawashdeh. An rtos-based run-time reconfigurable avionics system for uavs. In *AIAA Infotech@Aerospace 2010*, 2010.

[41] Michael Aiello, John Berryman, Jonathan Grohs, and John Schierman. Run-time assurance for advanced flight-critical control systems*. In *AIAA Guidance, Navigation, and Control Conference*, 2010.

[42] Edmond Wong, John D. Schierman, Thomas Schlapkohl, and Amy Chicatelli. Towards run-time assurance of advanced propulsion algorithms. In *50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, 2014.

[43] Remus Avram, Xiaodong Zhang, Jonathan A. Muse, and Matthew Clark. Nonlinear adaptive control of quadrotor uavs with run-time safety assurance. In *AIAA Guidance, Navigation, and Control Conference*, 2017.

[44] S. Berezin. Model checking and theorem proving: A unified framework (doctoral dissertation). school of computer science. *Carnegie Mellon University. Pittsburgh, PA*, 2002.

[45] Matt Webster, Neil Cameron, Michael Jump, and Michael Fisher. Towards certification of autonomous unmanned aircraft using formal model checking and simulation. In *Infotech@Aerospace 2012*, 2012.

[46] Nathan Good, Omar Aboutalib, Bea Thai, Neil Yamaoka, Charles Kim, Colin Wilkinson, and David Findlay. Validation process of the physics-based modeling of navigation sensors for sea-based aviation automated landing. In *AIAA Modeling and Simulation Technologies Conference*, 2016.

[47] Marco Bakera, Tiziana Margaria, Clemens D. Renner, and Bernhard Steffen. Game-based model checking for reliable autonomy in space. *Journal of Aerospace Computing, Information, and Communication*, 8(4):100–114, 2011.

[48] Gopinadh Sirigineedi, Antonios Tsourdos, Brian White, and Rafal Zbikowski. Kripke modelling and model checking of a multiple uav system monitoring road network. In *AIAA Guidance, Navigation, and Control Conference*, 2010.

[49] Laura Humphrey. Model checking uav mission plans. In *AIAA Modeling and Simulation Technologies Conference*, 2012.

[50] Giovanni Verzino. Model checking driven simulation of sat procedures. In *AIAA SpaceOps 2012 Conference*, 2012.

[51] Laura Humphrey and Michael Patzek. Model checking human uav mission plans. In *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2013.

[52] Christoph Torens and Florian Adolf. Using formal requirements and model-checking for verification and validation of an unmanned rotorcraft. In *AIAA Infotech @ Aerospace*, 2015.

[53] Jeffery P. Hansen and Lutz Wrage. Verification of real-time systems using statistical model checking. In *AIAA Infotech @ Aerospace*, 2015.

[54] Jayaprakash Suraj Nandiganahalli, Sangjin Lee, and Inseok Hwang. Flight deck mode confusion detection using intent-based probabilistic model checking. In *AIAA Information Systems-AIAA Infotech @ Aerospace*, 2017.

[55] M. Ouimet. *Formal Software Verification: Model Checking and Theorem Proving. Technical Report ESL-TIK-00214*. Embedded Systems Laboratory, Massachusetts Institute of Technology, 2008.

[56] G. Sutcliffe, E. Denney, and B. Fischer. Practical proof checking for program certification. Retrieved on 19 December 2017 from https://ti.arc.nasa.gov/m/profile/edenney/papers/escar05.pdf, 2005.

[57] César Muñoz, Aaron Dutle, Anthony Narkawicz, and Jason Upchurch. Unmanned aircraft aystems in the national airspace system: A formal methods perspective. *ACM SIGLOG News*, 3(3):67–76, 2016.

[58] Alwyn Goodloe, Carl Gunter, and Mark-Oliver Stehr. Formal prototyping in early stages of protocol design. In *Proceedings of the 2005 Workshop on Issues in the Theory of Scurity*, 2005.

[59] Ying Jiang, Jian Liu, Gilles Dowek, and Kailiang Ji. Sctl: Towards combining model checking and proof checking, 2016.

[60] S. Asokan, G. S. Kumar, and N. J. Lal. Modeling of alfa programs using pvs theorem prover. pages 373–375. IEEE, 2009.

[61] C Muñoz. Formal methods in air traffic management: The case of unmanned aircraft systems (invited lecture), lecture notes in computer science, vol 9399, 58-62. In *Proceedings of the 12th International Colloquium on Theoretical Aspects of Computing*, 2015.

[62] Yazdi I. Jenie, Erik-Jan Van Kampen, Joost Ellerbroek, and Jacco M. Hoekstra. Safety assessment of unmanned aerial vehicle operations in an integrated airspace. In *AIAA Infotech @ Aerospace*, 2016.

[63] Sergio Guarro, Michael K. Yau, Umit Ozguner, Tunc Aldemir, Arda Kurt, Mohammad Hejase, and Matt Knudson. Formal framework and models for validation and verification of software-intensive aerospace systems. In *AIAA Information Systems-AIAA Infotech @ Aerospace*, 2017.

[64] Steve Chien, Joshua Doubleday, David R. Thompson, Kiri L. Wagstaff, John Bellardo, Craig Francis, Eric Baumgarten, Austin Williams, Edmund Yee, Eric Stanton, and Jordi Piug-Suari. Onboard autonomy on the intelligent payload experiment cubesat mission. *Journal of Aerospace Information Systems*, 14(6):307–315, 2017.

[65] Caris Moses, Rahul Chipalkatty, and Robert Platt. Belief space hierarchical planning in the now for unmanned aerial vehicles. In *AIAA Infotech @ Aerospace*, 2016.

[66] Christian Carreon-Limones, Andrew Rashid, Phillip Chung, and Subodh Bhandari. 3-d mapping using lidar and autonomous unmanned aerial vehicle. In *AIAA Information Systems-AIAA Infotech @ Aerospace*, 2017.

[67] Pooja Agrawal, Ashwini Ratnoo, and Debasish Ghose. Image segmentation-based unmanned aerial vehicle safe navigation. *Journal of Aerospace Information Systems*, 14(7):391–410, 2017.

[68] Haiyang Chao, Yu Gu, Jason Gross, Matthew Rhudy, and Marcello Napolitano. Flight-test evaluation of navigation information in wide-field optical flow. *Journal of Aerospace Information Systems*, 13(11):419–432, 2016.

[69] Alok Desai and Dah-Jye Lee. Efficient feature descriptor for unmanned aerial vehicle ground moving object tracking. *Journal of Aerospace Information Systems*, 14(6):345–350, 2017.

[70] Vincent E. Roback, Diego F. Pierrottet, Farzin Amzajerdian, Bruce W. Barnes, Alexander E. Bulyshev, Glenn D. Hines, Larry B. Petway, Paul F. Brewster, and Kevin S. Kempton. Lidar sensor performance in closed-loop flight testing of the morpheus rocket-propelled lander to a lunar-like hazard field. In *AIAA Guidance, Navigation, and Control Conference*, 2015.

[71] Raghvendra V. Cowlagi and Joseph Sperry. Unifying artificial intelligence and trajectory optimization for uav guidance. In *AIAA Guidance, Navigation, and Control Conference*, 2016.

[72] Cesar Muñoz and Anthony Narkawicz. Formal analysis of extended well-clear boundaries for unmanned aircraft. Hampton, 2016. NASA Center for AeroSpace Information (CASI). Conference Proceedings.

[73] R. W. Ghatas, D. P. Jack, D. Tsakpinis, M. J. Vincent, J. L. Sturdy, C. A. Muñoz, K. D. Hoffler, A. M. Dutle, R. Myer, A. M. DeHaven, T. Lewis, and K. E. Arthur. Unmanned aircraft systems minimum operational performance standards end-to-end verificaion and validation (e2-v2) simulation. In *Technical Memorandum, NASA/TM-2017-20780*, 2017.

[74] Anthony Narkawicz, Cesar Munoz, and Aaron Dutle. Coordination logic for repulsive resolution maneuvers. In *16th AIAA Aviation Technology, Integration, and Operations Conference*, 2016.

[75] Subodh Bhandari and Jonathan Novak. Neural network based control of an airplane uav using radial basis functions. In *AIAA Guidance, Navigation, and Control Conference*, 2015.

[76] D. H. Costello and H. Xu. Generating certificaion evidence for autonomous aerial vehicles decision-making. Accepted for Publication 20 September 2020 in the AIAA Journal of Aerospace Information Systems, 2020.

[77] Mica R. Endsley. Design and evaluation for situation awareness enhancement. *Human Factors and Ergonomics Society Annual Meeting Proceedings*, 32(2):97–101, 1988.

[78] Chris Eldridge. Electronic eyes for the allies: Anglo-american cooperation on radar development during world war ii. *History and Technology*, 17(1):1–20, 2000.

[79] M. Bradley, M. Nealiegh, J.S. Oh, and P. Rothberg. Combat casualty care and lessons learned from the last 100 years of war. *Current Problems in Surgery*, 2017.

[80] Hermione Giffard. Engines of desperation: Jet engines, production and new weapons in the third reich. *Journal of Contemporary History*, 2013.

[81] S. A. Rodd. *Government Laboratory Technology Transfer: Process and Impact Assessment*. PhD thesis, Virginia Polytechnic Institute and State University, 1998.

[82] C.A. Alexander. Development of the combiner-eyepiece night-vision goggle. In *1990 Technical Symposium on Optics, Electro-Optics, and Sensors*, 1990.

[83] M. Kelly. Ea-18g in flight test over maryland. From the private collection of CDR Costello, 2009.

[84] Certification of percision approach landing systems on aircraft carriers, amphibious assult ships and naval air stations. Department of the Navy, Naval Air Systems Command, NAVAIR Instruction 13800.19, 2016.

[85] G.M. Kurtz. Certification of percision approach landing systems on aircraft carriers, amphibious assult ships and naval air stations manual. Department of the Navy, Naval Air Systems Command, NAVAIR M-13800.1, 2016.

[86] J Jolly and C. Clark. Cvn precision approach and landing systems (pals) certification tests. NAVAIR, Test Plan Number SA17-02-0721B02, 2018.

[87] R. Scherman. Cvn-68 nimitz-class. Retrieved 3 December 2017 from https://fas.org/man/dod-101/sys/ship/cvn-68.htm.

[88] R. McCallum. Star wars: Episode iii revenge of the sith, 2011.

[89] N. Subbaraman. X-27b navy dorne completes first ever unmanned carrier landing. Retrieved 5 December 2017 from https://www.nbcnews.com/technology/x-47b-navy-drone-take-first-stab-unmanned-carrier-landing-6C10591335.

[90] Jace L. Allen. An overview of model-based development verification/validation processes and technologies in the aerospace industry. In *AIAA Modeling and Simulation Technologies Conference*, 2016.

[91] C. Fields. Report of the defense science board summer study on autonomy. Technical report, Defense Science Board, Department of Defense, 2016.

[92] Clay J. Humphreys, Richard Cobb, David R. Jacques, and Jonah A. Reeger. Dynamic re-plan of the loyal wingman optimal control problem. In *AIAA Guidance, Navigation, and Control Conference*, 2016.

[93] M. B. Donley and N. A. Schwartz. Technology horizons, a vision for air force science and technology 2010-30. Technical report, Office of the US Air Force Chief Scientist, 2010.

[94] D.M. Tate, R.A. Grier, C.A. Martin, F. L. Moses, D. A. Sparrow, J. R. Edmonson, S. Chaki, D. H. Scheidt, D. H. Scheidt, C. D. Piatko, D. Davis, and

D. Stausberger. A framework for evidence-based licensure of adaptive autonomous systems: Technical areas. Technical report, Institute for Defense Analyses, IDA Paper P-5325, Log H 16-000680, 2016.

[95] Kevin A. Hoff and Masooda Bashir. Trust in automation: Integrating empirical evidence on factors that influence trust. *Human Factors: The Journal of Human Factors and Ergonomics Society*, 57(3):407–434, 2015.

[96] Cheolhyeon Kwon, Scott Yantek, and Inseok Hwang. Real-time safety assessment of unmanned aircraft systems against stealthy cyber attacks. *Journal of Aerospace Information Systems*, 13(1):27–45, 2016.

[97] Usaf airworthiness af162-601. United States Air Force, 2010.

[98] Matthew Dillsaver, Matthew Clark, and Xiaodong Zhang. Military airworthiness certification of autonomous air vehicles with adaptive controllers. In *AIAA Information Systems-AIAA Infotech @ Aerospace*, 2017.

[99] M Morris, Dean and Kevis MacG. Adams. The hole is more than the sum of its parts: Understanding and managing emergent behavior in complex systems. Software Project Management - Lessons Learned, January/Feburary 2013.

[100] John Penn. Testability, test automation and test driven development for the trick simulation toolkit. In *AIAA Modeling and Simulation Technologies Conference*, 2014.

[101] Jimmy E. Rico and Kamran Turkoglu. Arduino based low-cost experimental unmanned aerial flight system for attitude determination in autonomous flights. In *AIAA Modeling and Simulation Technologies Conference*, 2016.

[102] Khalil Ghorbal, Jean-Baptiste Jeannin, Erik Zawadzki, André Platzer, Geoffrey J. Gordon, and Peter Capell. Hybrid theorem proving of aerospace systems: Applications and challenges. *Journal of Aerospace Information Systems*, 11(10):702–713, 2014.

[103] Pierre Courtieu, Lionel Rieg, Sébastien Tixeuil, and Xavier Urbain. Impossibility of gathering, a certification. *Information Processing Letters*, 115(3):447–452, 2015.

[104] Anthony Narkawicz and Cesar Muñoz. A formally verified conflict detection algorithm for polynomial trajectories. In *AIAA Infotech @ Aerospace*, 2015.

[105] A. Narkawicz and C. Muñoz. Formal verificaion of conflict detection algorithms for arbitray trajectories. *Reliable Computing 17(2). Retrieved 19 Jan 2020 from, https://interval.louisiana.edu/reliable-computing-journal/volume-17/reliable-computing-17-pp-2. pp. 09-237.pdf*, June 2012.

[106] Hu Huang, Samuel Guyer, and Jason Rife. Applying machine learning for run-time bug detection in aviation software. In *AIAA Infotech @ Aerospace*, 2016.

[107] Matthew Dillsaver, Matthew Clark, and Xiaodong Zhang. Military airworthiness certification of autonomous air vehicles with adaptive controllers. In *AIAA Information Systems-AIAA Infotech @ Aerospace*, 2017.

[108] Shankar Sankararaman. Towards a computational framework for autonomous decision-making in unmanned aerial vehicles. In *AIAA Information Systems-AIAA Infotech @ Aerospace*, 2017.

[109] Nicholas Sweet, Nisar R. Ahmed, Ugur Kuter, and Christopher Miller. Towards self-confidence in autonomous systems. In *AIAA Infotech @ Aerospace*, 2016.

[110] Chimpalthradi R. Ashokkumar and George W. York. Trajectory transcriptions for potential autonomy features in uav maneuvers. In *AIAA Guidance, Navigation, and Control Conference*, 2015.

[111] Hao Lyu, Jayaprakash Suraj Nandiganahalli, and Inseok Hwang. Human automation interaction issue detection using a generalized fuzzy hidden markov model. In *AIAA Information Systems-AIAA Infotech @ Aerospace*, 2017.

[112] Guillaume Brat. Reducing v and v cost of flight critical systems: Myth or reality. In *AIAA Information Systems-AIAA Infotech @ Aerospace*, 2017.

[113] D.M. Tate, R.A. Grier, C.A. Martin, F.L. Moses, and D. Sparrow. A framework for evidence-based licensure of adaptive autonomous systems. Technical report, Institute for Defense Analyses, IDA Paper P-5325, Log: H 16-000084, 2016.

[114] Federico Garcia Lorca, Srikanth Gururajan, and Stephen Belt. Characterization of pilot profiles through non-parametric classification of flight data. In *AIAA Information Systems-AIAA Infotech @ Aerospace*, 2017.

[115] Jaime L. Junell, Erik-Jan Van Kampen, Coen C. de Visser, and Q. Ping Chu. Reinforcement learning applied to a quadrotor guidance law in autonomous flight. In *AIAA Guidance, Navigation, and Control Conference*, 2015.

[116] Ashraf Kamal, Amgad M. Aly, and Ahmed Elshabka. Tuning of airplane flight dynamic model using flight testing. In *AIAA Modeling and Simulation Technologies Conference*, 2015.

[117] G. C. Wilson. Eject! eject! eject!, 1993.

[118] G. E. P. Box. Robustness in the strategy of scientific model building. Wisconsin Univ-Madison Mathematics Research Center, 1979.

[119] Jared K. Wikle, Timothy W. McLain, Randal W. Beard, and Laith R. Sahawneh. Minimum required detection range for detect and avoid of unmanned aircraft systems. *Journal of Aerospace Information Systems*, 14(7):351–372, 2017.

[120] D.S. Young. Safe for use certification for ground based sense and avoid (gb-saa) systems, marine corps air station (mcas) cherry point, north carolina in support of vmu-2 rq21a and rq-7b operations. NAVAIR Ser Air-4.1/16-016, 2016.

[121] Subodh Bhandari, Jeremiah Farinella, and Clayton Lay. Uav collision avoidance using a predictive rapidly-exploring random tree (rrt). In *AIAA Infotech @ Aerospace*, 2016.

[122] Hanseob Lee, Dasol Lee, and David Hyunchul Shim. Receding horizon-based rrt* algorithm for a uav real-time path planner. In *AIAA Information Systems-AIAA Infotech @ Aerospace*, 2017.

[123] R.D. McFadden. Pilot is hailed after jetliner's icy plunge, 2009.

[124] Julian P. McCafferty, Davis A. Woodward, Graham Ray, Amir Bachelani, and Bella Kim. Investigation of an autonomous landing sensor for unmanned aerial systems. In *AIAA Guidance, Navigation, and Control Conference*, 2014.

[125] Natops flight manual, mh-60r helicopter. Chief of Naval Operations and Under the Direction of the Commander, Naval Air Systems Command, 2014.

[126] Preliminary natops flight manual, mq-8b/c unmanned aircraft system. Chief of Naval Operations and Under the Direction of the Commander, Naval Air Systems Command, 2019.

[127] Eladio Domı´nguez, Beatriz Pérez amd Ángel L. Rubio, and Marı´a A. Zapata. A systematic review of code generation proposals from state machine specifications. *Information and Software Technology*, 54(10), 2012.

[128] Mouza Al Blooshi, Shafer Jafer, and Krishan Patel. Review of formal agile methods as cost-effective airworthiness certification processes. *Journal of Aerospace Information Systems*, 15(8):471–484, 2018.

[129] S. Kumar, R. S. Suryavanshi, and G. Chandra. Formal methods: Techniques and languages for software development. *Intl J. Enff Sci Advance Research*, 1(1):35–42, March 2015.

[130] D.N. Hoover, D. Gauspari, and P Humenn. Applications of formal methods to specification and safety of avionics software. Contractor Report (CR) NASA Document ID 19960023949, Odyssey Research Associates, Inc. Ithaca, NY United States, 1996.

[131] Judy Crow, Sam Owre, John Rushby, Natarajan Shankar, and Mandayam Srivas. A tutorial introduction to pvs. WIFT, 1995.

[132] Deliverables for contract n00014-12-c-0671, aurora flight sciences, 2018. Released by NAVAIR Center for Autonomy.

[133] Philip Koopman and Michael Wagner. Challenges in autonomous vehicle testing and validation. *SAE International journal of transportation safety*, 4(1):15–24, 2016.

[134] Sae international announces formation of new standards committee for artificial intelligence in aviation systems, 2019. Contify Aviation News.

[135] J. Jewell. Autonomy experimentation in an operationally relevant scenario. In *Vertical Flight Society's 75th Annual Forum and Technology Display*, 2019.

[136] J. Jewell. What's it doing now? testing autonomy on a manned planform. In *Society of Experimental Test Pilots 2018 East Coast Symposium*, 2018.

[137] Matthew S. Whalley, Marc D. Takahashi, Jay W. Fletcher, Ernesto Moralez III, Carl R. Ott, Michael G. Olmstead, James C. Savage, Chad L. Goerzen, Gregory J. Schulein, Hoyt N. Burns, and Bill Conrad. Autonomous black hawk in flight: Obstacle field navigation and landing-site selection on the rascal juh-60a. *Journal of Field Robotics*, 31(4), 2014.

[138] W. Duffie. Autonomous flight technology to provide rapid resupply for marines, 2017.

[139] J Grimes. Images from uh-1 over nas patuxent river taken from the pilots prespective at 200 ft agl. From the private collection of CDR Costello, 2020.

[140] J Weakley. Images from uh-1 over nas patuxent river taken from the pilots prespective at 200 ft agl. From the private collection of CDR Costello, 2020.

[141] D Costello. Images from aacus demonstrations. From the private collection of CDR Costello, 2017.

[142] W.G. Vincenti. What engineers know and how they know it: Analytial studies from aeronautical history. Johns Hopkins Univesity Press, 1990.

[143] R. J. Niewoehner, J.C. O'Conner, and R. Traven. What were learning about learning: Flight test implications. 61st Symposium, Society of Experimental Test Pilots, 2017.

[144] D. H. Costello, H. Xu, and J. Jewell. Autonomous flight test data in support of a safety of flight certification. Accepted for Publication 15 October 2020 in the AIAA Journal of Air Transport Submitted to AIAA, 2020.

[145] Mica R. Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors*, 37(1):32–64, 1995.

[146] T. Nguyen, C. P. Lim, N. D. Nguyen, L. Gordon-Brown, and S. Nahavandi. A review of situation awareness assessment approaches in aviation environments. *IEEE Systems Journal*, 13(3):3590–3603, 2019.

[147] Christopher D. Wickens, Juliana Goh, John Helleberg, William J. Horrey, and Donald A. Talleur. Attentional models of multitask pilot performance using advanced display technology. *Human Factors*, 45(3):360–380, Fall 2003. Copyright - Copyright Human Factors and Ergonomics Society Fall 2003; Last updated - 2017-11-10; CODEN - HUFAA6.

[148] Mica R. Endsley. Situation awareness misconceptions and misunderstandings. *Journal of Cognitive Engineering and Decision Making*, 9(1):4–32, 2015.

[149] Michael P. Snow and John M. Reising. Comparison of two situation awareness metrics: Sagat and sa-sword. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 44(13):49–52, 2016.

[150] Thomas Frey. Cooperative sensor resource management for improved situation awareness. In *Infotech@Aerospace 2012*, 2012.

[151] Kasey Ackerman, Enric Xargay, Donald A. Talleur, Ronald S. Carbonari, Alex Kirlik, Naira Hovakimyan, Irene M. Gregory, Christine M. Belcastro, Anna Trujillo, and Benjamin D. Seefeldt. Flight envelope information-augmented display for enhanced pilot situational awareness. In *AIAA Infotech @ Aerospace*, 2015.

[152] John Hogan, Eytan Pollak, and Mark Falash. Enhanced situational awareness for uav operations over hostile terrain. In *1st UAV Conference*, 2002.

[153] Holger Jaenisch, James Handley, and Louis Bonham. Data modeling for unmanned vehicle situational awareness. In *2nd AIAA "Unmanned Unlimited" Conf. and Workshop & Exhibit*, 2003.

[154] Karl Reichard, Jeff Banks, Eddie Crow, and Lora Weiss. Intelligent self-situational awareness for increased autonomy, reduced operational risk, and improved capability. In *1st Space Exploration Conference: Continuing the Voyage of Discovery*, 2005.

[155] Prasanna Velagapudi, Sean Owens, Paul Scerri, Michael Lewis, and Katia Sycara. Environmental factors affecting situation awareness in unmanned aerial vehicles. In *AIAA Infotech@Aerospace Conference*, 2009.

[156] Rashaad Jones, Laura Strater, Jennifer Riley, Erik Connors, and Mica Endsley. Assessing automation for aviation personnel using a predictive model of situation awareness. In *AIAA SPACE 2009 Conference & Exposition*, 2009.

[157] Alec J. Bateman, Michael DeVore, Nathan D. Richards, and Stephan De Wekker. Onboard turbulence recognition system for improved uas operator situational awareness. In *AIAA Scitech 2020 Forum*, 2020.

[158] Mica R. Endsley. From here to autonomy: Lessons learned from human–automation research. *Human Factors: The Journal of Human Factors and Ergonomics Society*, 59(1):5–27, 2017.

[159] Susana Ruano, Carlos Cuevas, Guillermo Gallego, and Narciso García. Augmented reality tool for the situational awareness improvement of uav operators. *Sensors (Basel, Switzerland)*, 17(2):297, 2017.

[160] Mark Hanson and Paul Gonsalves. Space situational awareness using intelligent agents. In *AIAA Space 2003 Conference & Exposition*, 2003.

[161] Jan-Christoph Scharringhausen, Andreas Kolbeck, and Thorsten Beck. A robot on the operator's chair - the fine line between automated routine operations and situational awareness. In *SpaceOps 2016 Conference*, 2016.

[162] Brian Collins, Lauren Kessler, and Elizabeth Benagh. An algorithm for enhanced situation awareness for trajectory performance management. In *AIAA Infotech@Aerospace 2010*, 2010.

[163] Christophe De Wagter and J.A. Mulder. Towards vision-based uav situation awareness. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005.

[164] David Shim and Shankar Sastry. A situation-aware flight control system design using real-time model predictive control for unmanned autonomous helicopters. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006.

[165] Bowen Lu, Matthew Coombes, Baibing Li, and Wen-Hua Chen. Aerodrome situational awareness of unmanned aircraft: An integrated self-learning approach with bayesian network semantic segmentation. *IET Intelligent Transport Systems*, 12(8):868–874, 2018.

[166] Lihua Zhu, Xianghong Cheng, and Fuh-Gwo Yuan. A 3d collision avoidance strategy for uav with physical constraints. *Measurement*, 77:40–49, 2016.

[167] Shixun You, Lipeng Gao, and Ming Diao. Real-time path planning based on the situation space of ucavs in a dynamic environment. *Microgravity Science and Technology*, 30(6):899–910, 2018.

[168] J.A. Adams. Unmanned vehicle situation awareness: A path forward. In *Human Systems Integration Symposium*, 2007.

[169] Christopher D. Wickens. Situation awareness and workload in aviation. *Current Directions in Psychological Science: A Journal of the American Psychological Society*, 11(4):128–133, 2002.

[170] G.E. Cooper and R. P. Harper. The use of pilot rating in the evaluation of aircraft handling qualities. NASA-TN-D-5143, Technical Report, 1969.

[171] Military specificaion: Flying qualities of piloted airplanes, mil-f-8785c, November 1980.

[172] John Hodgkinson. *Aircraft Handling Qualities.* AIAA Education Series, Reston, Virginia, 1999.

[173] Douglas C. Montgomery. *Design and analysis of experiments.* Wiley, New York, fourth edition, 1997.

[174] Z.A. McCarley and T.R. Jorris. Design of experiments used to investigate an f/a-18 e/f strafing aunomaly. In *11th AIAA SoCal Aerospace Systems & Technology (ASAT) Conference*, 2014.

[175] J.C. O'Conner. Introduction to aircraft and systems test and evaluation: Design of experiments screening/characterizing example. United States Naval Test Pilot School, SYS Short Course Module SY04, 2020.

[176] Peter D. Clive, Jeffrey A. Johnson, Michael J. Moss, James M. Zeh, Brian M. Birkmire, and Douglas D. Hodson. Advanced framework for simulation, integration and modeling (afsim) (case number: 88abw-2015-2258), 2015.

[177] Nicholas Hanlon, Eloy Garcia, David Casbeer, and Meir Pachter. Afsim implementation and simulation of the active target defense differential game. In *2018 AIAA Guidance, Navigation, and Control Conference*, 2018.

[178] Multiservice brevity codes. Air Land Sea Applaction Center, FM 3-987.18, MCRP 3-25B, NTTP 6-02.1, AFTTP(I) 3-2.5, Feburary 2002.

[179] Radar techniques - primer principles, april 1945 qst article, 2019.

[180] B. M. Dickerson. Navair public release 2020-643. NAWCAD, 2020.

[181] Jr Hair, Joe F., Marko Sarstedt, Lucas Hopkins, and Volker G. Kuppelwieser. Partial least squares structural equation modeling (pls-sem): An emerging tool in business research. *European Business Review*, 26(2):106–121, 2014. Copyright - Copyright Emerald Group Publishing Limited 2014; Last updated - 2019-09-06.

[182] D. H. Costello and H. Xu. Relating sensor degradation to vehicle situational awareness for autonomous air vehicle certification. Submitted to AIAA Journal of Aerospace Information Systems, September 2020, 2020.