DISSERTATION

AUTONOMOUS UAV CONTROL AND TESTING METHODS UTILIZING PARTIALLY OBSERVABLE MARKOV DECISION PROCESSES

Submitted by

Christopher M. Eaton

Walter Scott, Jr. College of Engineering

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Spring 2018

Doctoral Committee:

Advisor: Edwin K.P. Chong
Co-Advisor: Anthony A. Maciejewski

Thomas Bradley
Peter Young

ABSTRACT

AUTONOMOUS UAV CONTROL AND TESTING METHODS UTILIZING PARTIALLY OBSERVABLE MARKOV DECISION PROCESSES

The explosion of Unmanned Aerial Vehicles (UAVs) and the rapid development of algorithms to support autonomous flight operations of UAVs has resulted in a diverse and complex set of requirements and capabilities. This dissertation provides an approach to effectively manage these autonomous UAVs, effectively and efficiently command these vehicles through their mission, and to verify and validate that the system meets requirements. A high level system architecture is proposed for implementation on any UAV. A Partially Observable Markov Decision Process algorithm for tracking moving targets is developed for fixed field of view sensors while providing an approach for more fuel efficient operations. Finally, an approach for testing autonomous algorithms and systems is proposed to enable efficient and effective test and evaluation to support verification and validation of autonomous system requirements.

ACKNOWLEDGEMENTS

I would like to thank my co-advisors Professors Edwin K.P. Chong and Anthony A. Maciejewski for their masterful guidance and support throughout my PhD. I would also like to thank Professors Thomas Bradley and Peter Young for their guidance and support as part of my Doctoral Committee. The insight of these professors enabled my success in numerous ways. I would also like to acknowledge my supervisors and co-workers at the 412th Test Wing at Edwards AFB, CA who supported me throughout this process and offered excellent advice and independent review of my efforts with an eye towards value to our mission. I especially want to acknowledge Mr. Patrick Zang, Mr. Douglas Wada, Mr. Artemio Cacanindin, Mr. Jason Bostjancic, Mr. Reagan Woolf, Dr. James Brownlow, Maj. Danny Riley, Capt. Justin Merrick, and Ms. Sonja Crowder for their support, review, discussions, and recommendations. I would also like to acknowledge Mr. Matt Clark, Maj (Dr.) Garrison Lindholm, and Dr. Derek Kingston from the Air Force Research Lab at Wright-Patterson AFB, OH for their recommendations and inputs throughout this process and our collaboration on the concept of Services Based Testing of Autonomy.

Finally, and foremost, I must acknowledge my amazing wife Dr. Jessica Eaton and my children Andrew and Kayla. I would not have been successful without their support and understanding during these past 4 years as I spent more time working on school than with them. I want to also acknowledge my Father and Mother-in-law Alfonso and Rosa Corral and Brother and Sister-in-law Antonio and Carolina Espitia. I can't thank them enough for their support of Jessica and our kids during my many weekends of studying and travel. To my brother Doug Eaton, who was my support, mentor, and role model during my undergraduate years, I wouldn't have made it without him. Finally, if it wasn't for my amazing super-wife Jessica for the continual encouragement while also managing the house and our kids with all their activities during my countless hours working on my PhD, I would not have been successful!

DEDICATION

I would like to dedicate this dissertation to my late parents, Wayne and Carol Eaton, my wife Dr. Jessica Eaton, and my kids Andrew and Kayla. My parents continual encouragement to think beyond our small town and follow my dreams has continued to push me to strive for more. My wife continues to challenge me to be better every day. And finally, to my kids, may they strive to find whatever makes them happy and pursue it with all their heart.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The use of unmanned aerial systems (UASs) in both the public and military environments is predicted to grow significantly. As the demand for UASs grows, the availability of more robust and capable vehicles that can perform multiple mission types will be needed. In the public sector, the demand will grow for UASs to be used for agriculture, forestry, and search and rescue missions. Militaries continue to demand more UAS capabilities for diverse operations around the world. Significant research has been performed and continues to progress in the areas of autonomous UAS control. A majority of the work focuses on subsets of UAS control: path planning, autonomy, small UAS controls, and sensors. Minimal work exists on a system-level problem of multiple-scenario UAS control for integrated systems. This paper provides a high-level modular system architecture definition that is modifiable across platform types and mission requirements. A review of the current research and employment of UAS capabilities is provided to evaluate the state of the capabilities required to enable the proposed architecture.

Significant development in path planning algorithms for unmanned aerial vehicles (UAVs) has been performed using numerous different methods. One such method, Partially Observable Markov Decision Processes (POMDP), has been used effectively for tracking fixed and moving targets. One limitation of those efforts has been the assumption that the UAVs could always see the targets, with a few unique exceptions, e.g., building obscuration. In reality, there will be times when a vehicle will not be able to observe a target due to constraints such as turn requirements or tracking multiple targets that are not within a single field of view (FOV). The POMDP formulation proposed in this paper is robust enough to handle those missed observations. Monte Carlo runs of 1000 iterations per configuration are run to provide statistical confidence in the performance of the algorithm. UAV altitude and sensor configuration are varied to show robustness across multiple configurations. A sensor with a limited FOV is assumed and changes in fixed look angle are evaluated. Changes in altitude provide results equivalent to changes in sensor window or focal length.

Results show that the POMDP algorithm is capable of tracking single and multiple moving targets successfully with limited FOV sensors across a range of conditions.

The ability to effectively track moving targets is a critical capability for future autonomous aircraft. While many methods have been developed for performing target tracking, minimal work has focused on fuel-efficient options to extend mission duration. The ability to tightly track a target is critical for certain missions; however, increased tracking errors can be accepted in certain scenarios to extend endurance. Partially Observable Markov Decision Processes (POMDPs) have been shown to be effective for tracking fixed and moving targets. This paper provides a fuel-efficient option that shows a 10% endurance increase with adequate target tracking. The algorithm provides tracking with a limited field of view fixed sensor that will have limited observations depending on mission requirements. The POMDP formulation proposed in this paper is robust enough to handle observations while also providing options for improved fuel efficiency. We perform 500 Monte Carlo simulations per configuration to provide statistical confidence in the performance of the algorithm.

The test and evaluation (T&E) of autonomous systems, that adequately supports the verification and validation (V&V) process, is a significant challenge facing the test community. The ability to quickly and reliably test autonomy is necessary to provide a consistent T&E, V&V (TEVV) capability. A safe, efficient, and cost effective test capability, regardless of autonomy or sensor capability, is required. Autonomy and sensor capabilities, referred to as services, can be integrated easily into small Unmanned Aircraft Systems (sUAS) of differing capabilities and complexities. An integrated open-source architecture, for both software and hardware, implemented on multiple sUAS of varying capabilities can provide a robust test capability for emerging autonomous behaviors. The inclusion of a run time assurance (RTA) common safety watchdog and a Live-Virtual-Constructive (LVC) capability provides a consistent, robust, and safe test capability/environment. The use of an open software and hardware architecture ensures cross-platform viability. These features will allow test teams to focus on the newly incorporated autonomy and sensor services, not on other ancillary capabilities and systems on the test vehicle. Testing of the services in this manner will enable a common TEVV approach, regardless of final platform integration while decreasing risk and accelerating the availability of autonomy services. Services-Based Testing of Autonomy (SBTA) provides a cost-effective and focused capability to test autonomous services, whether software, hardware, or both.

CHAPTER 2

MULTIPLE-SCENARIO UNMANNED AERIAL SYSTEM CONTROL: A SYSTEMS ENGINEERING APPROACH AND REVIEW OF EXISTING CONTROL METHODS

2.1 INTRODUCTION

In July 2014, the Teal group predicted that worldwide Unmanned Aerial Systems (UASs) expenditures will grow to over $11 Billion per year with a total investment of over $91 Billion by 2024. It is expected that 86% of the market will be military and 14% will be in the civil market [1, 2]. As the growth continues, challenges and expectations will continue to rise as users will expect more robust and capable vehicles. The military market continues to expand and develop new capabilities and requirements. The growth in the civil market is expected to significantly expand as the rules on civil use of UASs in the US and around the world become better defined. As these markets expand, the need to have systems that can adapt to new missions, sensors, and environments will drive requirements.

The Defense Advanced Research Projects Agency (DARPA) released a Broad Agency Announcement (BAA) for the Collaborative Operations in a Denied Environment (CODE) Program in 2014 [3]. This BAA defines numerous requirements and expectations for future system capability of unmanned and autonomous vehicles working as single systems and multiple vehicle teams. Many of the requirements defined in the CODE BAA can be utilized to define system architecture and capabilities for both military and civilian systems. These requirements will provide a significant portion of the requirements for the system defined herein. In early 2014, DARPA also released a BAA for Distributed Battlespace Management (DBM) that proposes a series of automated and autonomous decision aids to assist battle managers and pilots [4]. The DBM envisions, amongst other needs, the ability to enable improved command and control of autonomous operations of UASs, including in manned-unmanned teams. There are multiple thrusts within the DBM program, but one of the primary ones is for improved distributed adaptive planning and control. Additionally, requirements that can be applied globally across UASs from the Federal Aviation Administration (FAA) and other federal, state, and local laws are considered.

There are a significant number of current and future uses for UASs throughout the military and civilian world. The military is currently using, and continues to anticipate increased usage of, these systems in numerous areas including intelligence, surveillance, and reconnaissance (ISR) [5, 6], data and communication interfaces [7], electronic warfare [8,9], and limited attack roles [10]. Future cargo and transport capabilities [11] along with search and rescue operations [12] have been envisioned. In the commercial world, there are almost limitless possibilities of uses. Currently, applications exist for agriculture, firefighting, police, sciences, and forestry [2]. Significant efforts in cargo delivery, data capabilities, search and rescue, and traffic information are underway. Evaluating all of these capabilities and needs result in four primary mission types: ISR, persistent loiter, delivery, and attack. These four mission types will define the needs and requirements of the majority of the future UASs across the industry.

An early systems engineering analysis to evaluating the requirements, needs, and capabilities must be performed in an attempt to define a system that can be robust and adaptive to current and future needs. Utilizing requirements defined in the CODE BAA and other resources, a set of requirements can be defined and utilized to develop a system architecture to meet the users' needs. Section 2.2 provides a high-level problem definition that is addressed by this system architecture design. Section 2.3 will provide this systems engineering review and architecture definition.

Future multiple-scenario capability will require the system to operate dynamically across one or more of the four mission types and numerous subsets of those missions. Multiple-scenario control algorithms and architectures will enable a single platform to perform multiple mission roles with minimal reconfiguration. A dynamic architecture that enables recognition of sensors, system capabilities, and requirements will ensure the platform enables multiple-scenario support.

Significant work in UAS control and autonomous processes have been performed. The system defined in this paper requires numerous capabilities to be matured, some that already exist and some that need significant development work. Section 2.4 provides a detailed review of the current state of existing methods and capabilities in UAS autonomous path planning and safety controls. Some of the work that has been completed needs some significant improvement to enable the transition of the capabilities from theory and lab environments to practical applications. Section 2.5 provides some recommended improvement areas and provides a focus for future planned work by the authors. Final conclusions are provided in Section 2.6.

2.2 PROBLEM DEFINITION

As the need for future autonomous flight grows and systems mature, a high-level framework of how to design and integrate autonomous systems into existing and new vehicles is needed. Work continues to be performed in developing control capabilities and algorithms required to enable autonomous flight. However, in order for a framework to work, it must provide an architecture that is open and easily modifiable across a diverse type of vehicles and sensor capabilities. The system must enable the autonomous system algorithms to run in a framework that enables maximum flexibility while understanding vehicle capabilities.

2.3 SYSTEM DESIGN

In order to ensure the ability of autonomous systems to function and be effective across multiple vehicle types, a framework needs to be defined that enables flexibility in design and functionality. A high-level systems engineering review of requirements has been performed based upon the CODE BAA [3] and DBM BAA [4] in Section 2.3.1. A system architecture has been defined that enables the flexibility of design and functionality for autonomous vehicles in Section 2.3.2. Current and future research needs are briefly discussed in Section 2.3.3.

2.3.1 SYSTEM REQUIREMENTS

The CODE BAA [3] identified four top-level goals for any system proposed and developed:

1. develop and demonstrate the value of collaborative autonomy in a tactical context;
2. rapidly transition the capability to the warfighter;
3. develop an enduring framework to expand the range of missions, platforms, and capabilities that can leverage collaborative autonomy; and
4. develop an open architecture that enables all members of the rich community of unmanned systems and autonomy researchers to contribute to current and future capabilities.

Similarly, the DBM BAA [4] identified a goal of adaptive planning and control that could be distributed across systems to aid a variety of vehicles, weapons, and sensors. The goal is to enable UASs to satisfy the commander's intent while operating in normal or limited communication environments. The ability to have an adaptive decision process across mission types is critical. Hierarchical task processing under limited communication will be an important enabler of autonomy. The autonomous capabilities should be vehicle agnostic. The UASs should be able to negotiate both high-level battle manager tasks and low-level tactical tasks. The capability will need to exist to execute cooperative tasks with other UASs and manned vehicles.

Seven key performance objectives were identified in the CODE BAA where significant improvements are sought and would be critical for any system also developed for the DBM BAA or any other project. Six of the objectives are discussed below. The seventh objective, transitionability is not considered in this review.

1. Mission Efficiency:

   Mission efficiency is an important requirement for both military and civilian operations. The cost of completing the mission needs to be considered. The expense of flying the vehicle along with the duration required to complete the mission are critical concerns for all parties involved. Additionally, the ability to quickly react to changes in mission requirements or system functionality is critical for robust systems of the future. The bulk of the review of current capabilities in existing systems will relate to mission efficiency and control of the system. Mission efficiency can be considered in countless manners, including time to complete the mission (time efficiency), fuel used to complete the mission (fuel efficiency), endurance of mission (endurance), and total number of tasks completed (task efficiency).

2. Communication requirements:

   Limited communication frequencies will be available in the future. Limited bandwidth and minimization of communication will be required in future operations and will be a feature for more autonomous vehicles. Additionally, communication in a denied electronic environment will necessitate limited communications. The ability to have communications in unique environments with cognitive capabilities will be required in the future [13].

3. Manning:

   Currently, the ratio of operators to vehicles is many-to-one but in the future the desire is to flip the ratio to be one-to-many. To support this change in operational manning, a significant increase in system autonomy must be created. A system must be able to automate its mission path and plan with minimal operator inputs. Hierarchical logic for decision making must be implemented to ensure the most effective completion of the mission and rapid response to mission or system changes. The system must be able to react to both external inputs (operator) and internal inputs (system and sensor data). Additionally, unique challenges of training and educating future operators will be critical [14]. The review of current capabilities of algorithms and autonomous features as it relates to mission efficiency will incorporate the considerations of reduction in manning of operations.

4. Command Station:

   Future command stations must be robust and provide significant situational awareness for the operator. Being able to command vehicles from a mobile or fixed-based control station will be necessary to ensure flexibility in capabilities. Interfaces that enable the operator to quickly upload new tasks and parameters will be necessary. Limited command station requirements and current capabilities will be addressed in this paper.

5. Openness of the architecture:

   Open system architecture is a key of any current and future system viability. To provide a system architecture that can be utilized across multiple system sizes and types it must employ an open architecture to minimize the costs of integration with any existing or new systems. The design proposed in this paper attempts to provide a framework architecture that would satisfy current open architecture standards. Limited review of open architecture requirements will be addressed in this paper.

6. Multi-mission capability:

   The ability of a system to perform multiple missions will be critical for future viability. Some airframes may not lend themselves to transition across the four primary mission types of ISR, loiter, delivery, and attack. However, a system that operates primarily in one or two mission areas should be able to perform multiple roles within those mission areas. For example, a system that has a primary role of ISR should be able to perform recurring observation of fixed targets but also be able to transition to tracking of moving targets or persistent observation over a fixed target. The ability of systems to perform multiple missions will be discussed throughout the paper.

2.3.2 SYSTEM ARCHITECTURE

In the early 2000s, Boskovich, *et al.* [15] defined a control architecture for decision making within an autonomous UAS framework. This architecture described a high-level framework for designing autonomous intelligent control systems for UASs. The architecture defined four layers of control: redundancy management, trajectory generation, path planning, and decision making. This general philosophy is still present in many of the design work prevalent today in UAS mission and path planning design. However, this architecture only considers the general control and path planning of a vehicle. To address the high-level requirements previously defined in Section 2.3.1, a system architecture needs to be developed that can provide the framework for system design and functionality for more than just the mission control of the vehicle. Additionally, a significant portion of the existing research in UAS path planning considers the vehicle a point mass and many of the approaches consider only constant altitude and airspeed. There is minimal consideration of actual vehicle dynamics in the existing research. In order to address the actual vehicle dynamics and utilize those dynamics to improve mission performance utilizing existing methods, an architecture that integrates vehicle dynamics and systems is needed.

Figure 2.1 provides the high-level architecture defined for our proposed autonomous system. The definition proposes five primary functions: Mission Management, Vehicle Management, Sensors Management, Communications Management, and Safety Management. These five areas provide the sufficient top-level framework for any system, regardless of mission and vehicle type. The advantage of this system is that it provides a modular functionality architecture that can be adjusted for specific vehicles but can be common across numerous vehicle types. The autonomous algorithms will reside within the mission management functionality and will be dependent upon common interfaces and architecture.

Figure 2.2 provides a lower-level definition of the system architecture with key critical functionalities within the primary management systems. The functions and capabilities within each management area could be changed depending on each vehicle. However, the interface to the mission management system needs to remain consistent. The key to the architecture is that each primary functional area has a controller that manages the overall function, but capabilities can be added or removed based upon mission and system requirements in a modular fashion without impacting the larger system. Additionally, depending on mission tasking and systems on board, the controller could enable or disable any resident capability to improve performance of mission objectives without changes to the software. The detailed description of each of the five primary functions and their subsidiary functions are provided below.

1. Mission Management:

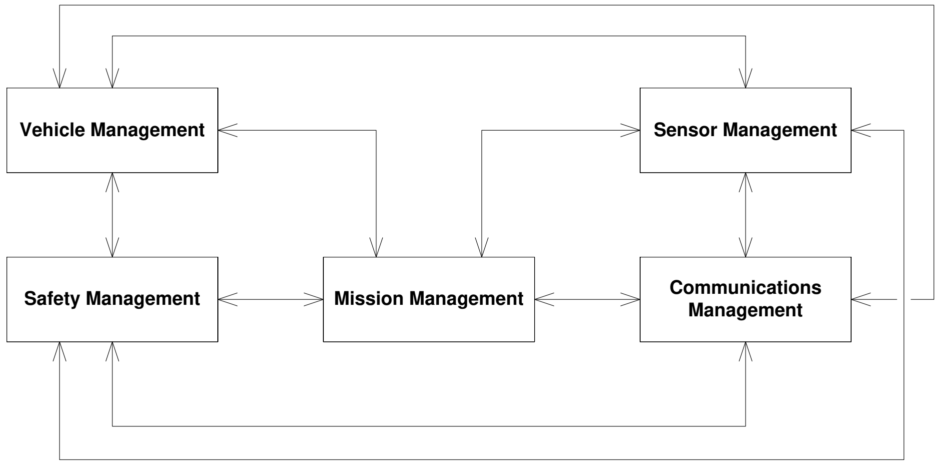

Figure 2.1: Top-Level System Architecture.

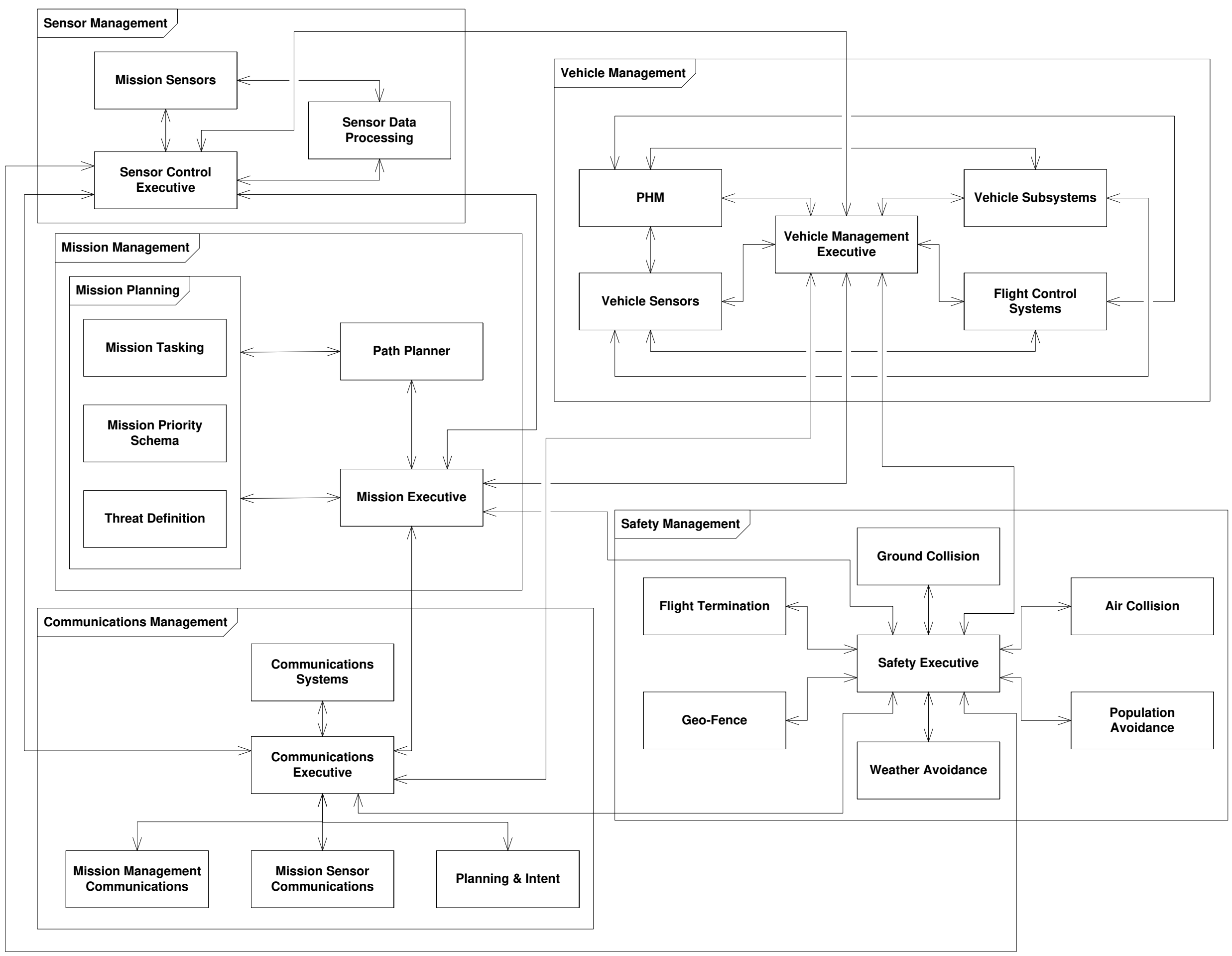

Figure 2.2: Detailed System Architecture.

The mission management function is the key to the success of the system architecture. A significant portion of the efforts in this area would be consistent with the work of Boskovic, *et al.* [15] as discussed earlier. The mission management will provide the primary high-level decision making for the mission performance of the vehicle. The mission management area is the focus of significant research in unmanned and autonomous control. There are three primary functions within the Mission Manager:

(a) Mission Planning:

The mission planning function provides the mission requirement details for the decision-making process of the mission executive. A mission-planning dataset could include definitions of tasks, priorities, and threats. The mission-planning dataset could be uploaded prior to a mission, during a mission, or self created depending on the autonomous capabilities designed within the system. The mission-tasking information would provide the required tasks the system is desired to perform. A mission-priority schema would provide the executive a decision framework to determine which task is of greater priority. For example, tracking a moving target could be defined as a higher priority than general reconnaissance data collection. Threat definition would provide the system considerations for areas to avoid due to known threats as well as considerations for how to handle newly discovered threats. These considerations could include keep-out zones, self protection actions with sensors, or other actions depending upon system capabilities.

(b) Path Planner:

The path planner is the key algorithm for defining where and how the vehicle should move. The path planner utilizes the considerations defined in the mission planning along with information provided (via the mission executive) on system states. The path planning algorithms could be dynamic or changed for any given mission based upon the needs, priority, and other considerations. The path planner must also determine path planning based upon contingency management requirements of the system for subsystem failures. The path-planning algorithm is a significant consideration of this paper and current capabilities are discussed later in this paper.

(c) Mission Executive:

The Mission Executive (ME) is the primary decision maker for the vehicle. The functionality of the ME defines whether to perform the current task defined by the path planner or reacts to safety management information. The ME also provides and receives communication updates with other vehicles, operators, and other sources as required. The ME commands the vehicle management system to perform flight maneuvers and other vehicle system functionality. The ME could be considered equivalent to a human operator within a manned vehicle system.

2. Sensor Management

Sensor Management provides the control for all the mission sensors installed on the vehicle. Mission sensors are defined as any sensor utilized to perform the mission. Sensors that are used to manage the vehicle control and health are handled within vehicle management. There may be some cross utilization of these sensors for both systems. However, the management of those sensors would be handled by their primary user. Portions of sensor management, such as sensor types and control, are well understood in existing systems. However, the sensor data processing will require continued and significant research to provide autonomous sensor data at a decision level that can be trusted. There are three key functions within the sensor management framework.

(a) Mission Sensors:

Mission sensors are the sensors specifically installed on the aircraft for data gathering in direct support of mission completion. The mission sensors will be dependent upon the vehicle and mission requirements. These sensors will perform the primary mission duties and could include electro-optical/infrared sensors, radar sensors, radio frequency sensors, or any number of other types. The sensors will have a direct interface to the data-processing module and the control-executive module. These sensors will perform their tasks based upon commands received from the sensor-control executive.

(b) Sensor-Data Processing:

The sensor-data processor will analyze received data and make a decision on the information received based upon algorithms defined. The processing could be utilized for any number of tasks including target recognition, geo-location, target motion, and sensor response. The data will also be processed for transmission, as required, and sent to the sensor-control executive for passage to the communication management for dissemination. A significant level of research is ongoing in areas of sensor data fusion, image processing, and recognition that can support decisions and vehicle tasking.

(c) Sensor Control Executive:

The sensor-control executive (SCE) is the primary controller of all sensors and sensor taskings. The SCE interfaces with the ME and provides sensor availability, sensor capability, sensor data evaluation (target ID, geo-location, *etc.*), and sensor health. The ME provides the SCE with sensor tasking. The SCE will be required to automatically identify what sensors it has installed on board and what their capabilities are.

3. Safety Management

Safety Management provides overall safety monitoring for the vehicle. The types of safety management performed can be dependent upon vehicle type, sensors installed, capabilities required, and vehicle capabilities. The systems defined in this architecture are notional but are critical for UASs. The safety executive can provide high-priority tasking to the mission executive that can result in overriding current activities for safety reasons. Safety management is an area that is understood, but the integration of it with autonomous systems continues to be researched and developed across vehicle types. The core safety management functions defined in this architecture are explained here, but are not exhaustive of possible functions.

(a) Safety Executive:

The Safety Executive (SE) processes all information from safety management capabilities and provides that information to the mission executive for execution. The SE will prioritize which safety feature should be addressed first (if multiple safety issues are occurring at the same time) and determine the recommended actions.

(b) Collision Avoidance:

Collision Avoidance algorithms for both ground collision and air-to-air would reside within the safety management area. These algorithms would determine when the vehicle is at risk of impacting something and provide recommended action(s) to avoid these problems.

(c) Flight Termination:

Flight Termination is a key issue for unmanned air vehicles. Flight termination can include destructive actions which result in the destruction of the vehicle. However, it can also contain contingency efforts that include immediate landing, reduction in system capabilities, flight-plan alteration, or other functionalities depending upon the mission and range requirements.

(d) Geo-Fence:

The geo-fence capability defines the areas within or outside of which a vehicle should maintain a presence. There may be unique mission requirements that require a system to fly in certain areas which the geo-fence may not allow based upon changes in mission or knowledge of areas of operation. If the vehicle is approaching a fence limit or has crossed a fence, the safety system should direct the vehicle back within the defined boundary. This geo-fence could be dynamic based upon known aircraft (air collision avoidance), major changes in weather (weather avoidance), or known threats and borders.

(e) Weather Avoidance:

Depending on vehicle capabilities or mission sensor capabilities and requirements there may be a need to avoid undesirable weather. Weather avoidance would provide keep-out areas to the safety executive that could be provided to either the mission executive or the geo-fence capability for management of vehicle path. For sensor functionality issues it would be more critical to provide that information to the mission executive for determination of path route and sensor tasking.

(f) Population Avoidance:

Mission requirements may require the vehicle to perform tasks in areas with significant or critical populations. As a result, there may be a need to fly close to population but avoid interference or impact with people and activities. The population avoidance functionality would determine where the vehicle needs to be to avoid the population of concern and provide the safety executive of how to react to the given situation.

4. Communications Management

Communications Management provides the key interface between the vehicle and other systems. The ability to send and receive both mission information and sensor data can be critical to the success of a given mission. By managing communications separately from the primary processes, it enables changes in communication methods without impacting the underlying functionality of the vehicle. Communications Management will divide the data as either mission management, mission sensor, and planning and intent. There are five primary functions within the Communications Manager:

(a) Communications Executive:

The communications executive provides the primary interface between the mission executive and the installed communication systems. The communication systems installed could vary depending on the vehicle type and mission requirements. The communication executive will provide external communications and data dissemination as required. The system will also need to recognize when communications are not being received for potential operations in a denied environment.

(b) Communications Systems:

The vehicle could have one or multiple communication systems installed for external communications capabilities dependent on mission requirements and vehicle capabilities. Communication types could include line of sight RF, satellite communications, optical/laser or others. Dissemination of data received and to be sent will be via the communications executive.

(c) Mission Management Communications:

Mission management communications will provide mission status, priority, tasking, and threat information for other vehicles. The communications executive will also update this information from any received data for processing via the mission executive.

(d) Mission Sensors Communications:

The mission sensor data will be processed and sent separately from other priority tasks (mission management, planning and intent) to provide external users with specific sensor data for analysis and use. By handling the mission sensor data separately from the other data, it prevents critical data being held up by sensor data dissemination. Mission and vehicle tasking data should take priority over data dissemination tasks. This separation will also enable a system to have separate communication systems for data and mission tasks.

(e) Planning and Intent:

The planning and intent data will provide current information on where the vehicle is, where it is going and the intents of its upcoming efforts. This will allow any other vehicles or operators in the mission to monitor and understand the plans of the vehicle. This information will enable users and vehicles to make decisions and recommendations on mission plans and efforts.

5. Vehicle Management

   The Vehicle Management system is responsible for the control of the vehicle and systems. The flight control system, vehicle subsystems, vehicle state (via Prognostics and Health Monitoring (PHM) and sensors) all reside within the vehicle management system. Vehicle management is well understood and is standard in most manned and unmanned aircraft. While there is significant research and work ongoing in this area, especially in the areas of PHM and fault tolerant operations, the underlying requirements and architecture are not significantly different from existing platforms. There are five primary areas within the vehicle management system.

   (a) Vehicle Management Executive:

       The Vehicle Management Executive (VME) manages the vehicle systems control and processing. The mission executive provides the tasking that the vehicle must perform and provide for processing. The data provided is then sent to the flight control systems, vehicle systems, and any other ancillary systems installed that require control. The VME also accepts sensor data and PHM data for processing and determination of whether degraded systems exist and if actions need to be taken. This data is also provided to the mission executive for mission tasking decisions.

   (b) Flight Control Systems:

       The Flight Control Systems (FCS) of a vehicle can include propulsion, flight control surfaces, flight control sensors, and any other system required for vehicle control. The FCS design and performance is unique to any given vehicle and needs to be provided to the path planner for determination of proper, efficient, and effective path planning.

   (c) Vehicle Subsystems:

       Vehicle subsystems can include ancillary systems such as electrical, hydraulic, environmental controls, and landing gear. These subsystems provide critical functionality that support the primary flight controls and mission sensors. Subsystems are generally well understood for existing areas but new and improved capabilities (especially in electrical power capabilities) continue to improve the state of these systems.

(d) Prognostics and Health Monitoring:

Prognostics and Health Monitoring (PHM) can provide an estimate of current and future health and capabilities of installed systems. Significant research has been performed and continues to be performed in this area. Fault tolerant design and functions also continue to be researched and can be integrated with PHM functionalities. PHM may or may not be present on a given vehicle but can provide enhanced control and insight into current and future performance.

(e) Vehicle Sensors:

Vehicle Sensors can be numerous and diverse across a vehicle. Depending on the size of the vehicle and criticality of the system there may be minimal or extensive sensing. The sensors can include critical flight data such as vehicle speed, rates, and accelerations via air data and/or inertial systems. Sensors can also perform pressure, temperature, voltage, or other critical measurements to support real-time performance or prognostics of future performance. Vehicle sensors continue to evolve and develop based upon new technology and needs.

2.3.3 SYSTEM NEEDS

Critical work continues to be performed in all areas of autonomous vehicle systems. The following discussions will provide details on current and ongoing work in selected areas. In the area of vehicle controls there continues to be significant work being performed on flight control systems based upon new and changing vehicle types and control schema. Fault-tolerant systems and PHM continue to be researched and system capabilities need to be improved for future autonomous system use. Sensor capabilities, sensor fusion, and sensor identification capabilities are areas that continue to be researched and will be critical for future use in autonomous systems. Mission management and path planning areas are seeing significant current research and will be required to be developed and become more effective for autonomous system use. Communication system capabilities continue to be researched for improved methods and needs, especially as the available frequency spectrum for communications is reduced for UAS applications. Safety management will continue to see research and growth as autonomy grows in order to improve and ensure trust of these autonomous systems.

2.4 REVIEW OF EXISTING METHODS AND CAPABILITIES

There are significant efforts ongoing in all areas of UAS autonomous control. This section will provide a review of the current state of path-planning and critical safety control as they relate to UAS autonomous controls. Section 2.4.1 provides a review of the current state of path planning algorithms that directly support the mission management capabilities defined in Section 2.3.2. Section 2.4.2 examines the state of critical safety control features that ensure safe flight, which directly support the safety management capabilities defined in Section 2.3.2.

2.4.1 PATH PLANNING

Path planning requires knowledge of target or mission needs in order to properly complete the planning algorithms. We will define three primary types of path planning: Fixed Target, Moving Target, and Target Search and Surveillance. Additionally, a system that can complete multiple scenarios within the three primary areas is valuable. An extension to multiple scenarios would be multiple aircraft supporting either the same type of mission or multiple scenarios.

Fixed Target

Fixed target path planning deals with the algorithms utilized with visiting fixed locations for information gathering or support. UASs are currently being used in missions that require the vehicle to visit a set of targets and maintain an optimum flight path to complete their tasks.

Many of the fixed target path planning problems can be considered similar to the traveling salesman problem (TSP) that has been evaluated significantly in numerous ways for decades since Dantzig *et al.*, developed a solution as an integer linear program [16]. The TSP type problem has been approached with multiple solutions. Many of the evolutionary algorithms used to solve these NP-hard (short for *non-deterministic polynomial-time hard*, widely taken to imply that the problem is computationally intractable [17]) problems provide acceptable results, although many of them have constrained the problems to some level.

Early approaches in solving the path planning problem included the use of a Tabu Search (TS) heuristic algorithm [18, 19]. The TS can provide a solution that allows for progression without becoming trapped in local optima. Ryan, *et al.* [20] used a Reactive Tabu Search method to solve UAS routing in the construct of a multiple Traveling Salesman Problem with time windows. Their objective was to maximize expected target coverage while incorporating weather and a survival probability at each target as random inputs. Wang *et al.* [21] proposed a Tabu (Taboo in their paper) Search algorithm for multiple task planning for multiple UASs that showed better performance than genetic algorithms or ant colony optimizations. Zhao and Zhao [22] utilized a Tabu search algorithm to develop their path and time. Numerous methods have been developed more recently that provide better results to the path-planning problem than the Tabu Search, although it still is valuable for certain processes. Additionally, TS algorithms are only useful for small problems that do not consider vehicle availability or tasks that require constraints beyond simple time ordering.

In 1956, Edsger Dijkstra proposed an algorithm for finding the shortest path between two points [23]. This algorithm is a key method for finding shortest paths for robots and unmanned vehicles and is used for numerous applications. The use of Dijkstra's algorithm can be found in countless applications to UAS path-planning applications [24–29].

Tong *et al.* [30] proposed a method of path planning that utilized Voronoi Diagrams and Discrete Particle Swarm Optimization (DPSO). A Voronoi diagram depicts lines that are equidistant to the closest neighboring points of interest, resulting in areas that define all points closest to the points of interest. A Voronoi diagram works similarly to and in conjunction with Dijkstra's algorithm. The lines from the Voronoi were used as an initial path, with the points of interest being threats to be avoided. A DPSO algorithm was then used for simultaneous target attacks by multiple vehicles.

Receding horizon control (RHC) [31] is a feedback control technique, also referred to as model predictive control, which is used across a large variety of applications. Receding horizon control is used as part of numerous algorithm types for UAS path planning. The advantage of RHCs is that it enables control of systems with a large number of inputs and outputs, especially for systems with complex objectives and strong nonlinear dynamics and constraints. The use of future considerations and predictions while optimizing the current time requirements is the key feature of RHC. Multiple path-planning algorithms utilize a form of RHC as part of their planning processes. The RHC control scheme has become useful for UAS algorithms due to the limited requirements for computational resources when compared to algorithms that perform global planning methods. Kuwata *et al.*, developed a decentralized RHC for multi-vehicle guidance [32, 33]. Xiao *et al.* [34] used an RHC method in conjunction with a virtual force method to improve the performance of the RHC. Peng *et al.* [35] developed a cooperative search algorithm utilizing RHC with a rapidly exploring random-tree path-planning algorithm. Schouwenaars *et al.* proposed a multiple aircraft trajectory planning algorithm utilizing a RHC strategy with a mixed integer linear programming basis [36]. There have been multiple efforts utilizing RHC methods in conjunction with Partially Observable Markov Decision Processes [27, 37–39].

In 1995, Kennedy and Eberhart [40] proposed a methodology of nonlinear function optimization using particle swarm optimization (PSO). This method provides a simple and computationally useful algorithm for optimizing a wide range of functions. The use of PSOs as part of a UAS path-planning algorithm has been employed successfully by numerous researchers [30, 35, 41–44]. Roberge *et al.* [45] provided a comparison of GAs and PSOs for UAS path planning. The resultant of the comparison shows that the GA produces superior trajectories to the PSO.

An approach that has shown good results is the use of Genetic Algorithms (GA). A GA provides a heuristic method based on natural evolution by defining the decision variable as a chromosome.

The chromosomes defined by the problem give the resultant population and an algorithm is utilized that generates an evolution process until a satisfactory solution results. Sahingoz [46, 47] and colleagues [48, 49] have performed significant work in using GAs for both single and multiple UAS path planning which has shown satisfactory results. Cheng *et al.* [50] developed an immune genetic algorithm that provided an ”immune operator and concentration mechanism” that improved convergence of existing GA algorithms. GAs can, under certain circumstances, suffer from premature convergence. Price and Lamont [51] used a GA design for self-organized search and attack of UAS swarms. Pehlivanoglu [52] proposed a vibrational GA algorithm enhanced with a Voronoi Diagram in an effort to improve the convergence problem. Research is continuing in the use of GAs for path planning of UASs [53, 54].

An algorithm based upon the annealing of metal [55,56] can be utilized to find global minimum of an objective. Drawing upon the annealing process, a Simulated Annealing (SA) algorithm will searchrandomly in the area of an initial guess. If an improvement is found, the new value is kept. If deterioration is noted, the result may be discarded or kept depending upon a temperature-dependent probability. A cooling schedule is used to determine when the temperature has been sufficiently cooled from the initial value. Turker *et al.* [57] presented a method for 2D path planning in a radar threat constrained environment using a simulated annealing algorithm. Leary *et al.* [26] evaluated five algorithms including SA, Consensus Based Bundle Algorithm (CBBA), greedy allocation, optimal Mixed Integer Linear Programming (MILP), and suboptimal MILP. The results showed that the SA algorithm provided the best solutions for path generation but required the longest computation time of the five algorithms; however, the growth in computation time with increased parameters was the lowest.

In 1992, Marco Doringo proposed an approach for finding an optimal path that drew upon the behavior of a colony of ants [58]. The ant colony optimization (ACO) approach has been adopted as a method to optimize UAS path planning. Fallahi *et al.* [59] proposed a method that integrated ACO and an analytic hierarchy process that showed good results for path planning using the ACO algorithm. An adaptive ant colony optimization approach for multiple UASs for coordinated trajectory re-planning was proposed by Duan *et al.* [60]. An extension of ACO looks more generically at digital pheromone responses and has been used to improve target search methods [61–63]. Shang *et al.* [64] proposed a hybrid algorithm that utilized GA and ACO algorithms for Multi-UAS mission planning which provided performance improvement over the two independent methods.

Moving Target

Moving Target path planning deals with the algorithms utilized to find and follow moving targets. Less work has been performed in the area of moving target tracking as compared to fixed target tracking. However, several similar algorithms to fixed target tracking, including Partially Observable Markov Decision Processes (POMDPs) and Genetic Algorithms (GAs), have been used with some success for finding and tracking moving targets.

Krishnamoorthy *et al.* [65, 66] developed a method of searching and tracking a moving target traveling with a known speed and direction on a road network while utilizing unattended ground sensors for target detection. This work was later developed and demonstrated with multiple-vehicles, multiple-targets, and a large series of ground sensors by Rasmussen and Kingston [67]. This approach relies upon unattended ground sensors to trigger when a moving target passes it's location. The sensor then informs the UAS of an intrusion and the associated information required to search and track the intruder. The system has shown some limitations due to sensor false alarms and delay in sending information due to limited line of sight data transmission capability. However, the functionality shows promise in supporting a network of ground sensors and vehicles to monitor roads or perimeters for intrusion.

Moon *et al.* [68] proposed the use of probability density functions in coordination with a negotiation task assignment framework for UAS tasking. The algorithm uses information gathering-based task assignment with a two-layer framework. An information-gathering layer uses the probability density functions to generate minimized value future trajectories. The task assignment layer utilizes a negotiation-based task allocation to assign tasks to the UASs in the network. Results showed promising results to search an area with minimal overlapping while finding all targets being searched.

Xiao *et al.* [34] proposed a virtual force and receding horizon method that enabled multiple-UAS cooperative search in a fixed region for unknown moving targets. Virtual force algorithm alone can be limited by being trapped at local minima while the receding horizon has large computational requirements that limits the length it can look ahead. The algorithm presented combined the two methods in order to alleviate the limitations of each method.

Sun and Liu [69] proposed a modified diffusion-based algorithm to manage target uncertainty while controlling multiple UASs with a hybrid receding horizon/potential method algorithm for a coordinated search for a moving target. The search area was divided into cells and the algorithm coordinated vehicle search tasks based upon weighting of cells of the search region. The cells not searched that were closer to a given UAS were given a higher weighting than ones closer to a different UAS. A hybrid method that combined potential and receding horizon methods was used to reduce the computational burden.

Frew *et al.* [70] and Summers *et al.* [71, 72] proposed similar control algorithms for multiple UAS coordinated standoff tracking of moving targets by utilizing Lyapunov guidance vector fields. Both approaches utilized Lyapunov guidance vector fields to generate stable paths for the UASs to fly while tracking a moving target. Multiple-UASs could be used by phasing them around the vector field solution. Both approaches showed acceptable results for multiple vehicles orbiting and tracking a moving target.

Geyer [73] proposed a method for urban searching of a moving target that considered complex geometry from buildings that can impact the ability of the sensor to see the target. The method utilizes search trees and particle filters to evaluate path options and provides efficient filtering along with a method of compressing the visibility function.

Bertuccelli and How [74] propose a Markov chain-like model for target motion estimation approach similar to particle filtering in order to account for the uncertainties in the target location estimates. Stochastic simulations of realizations of the transition matrix with posterior distribution approximation enable easy re-sampling of the posterior distribution. This method is valuable for searching for moving targets where the models of the target motion are poorly known.

Ragi and Chong [39, 75] proposed a method of UAS control utilizing POMDPs for tracking moving targets including evasive targets and threat avoidance. The resulting algorithm enabled a vehicle, or multiple vehicles, to be able to track moving targets. The design was robust enough to be able to track an evasive ground vehicle as well as avoid threats, obstacles, and other friendly vehicles while maintaining tracking of the target. Wind compensation and variable speed and altitude capabilities were integrated as well.

Target Search and Surveillance

Target search path planning deals with searching for targets with no or minimal information on the target of concern. Surveillance deals with repeated coverage and search of a specified area to obtain the desired information. Search problems are generally defined by generating a grid of cells over an environment. Poor information about target locations and noisy sensors can increase the difficulty of quickly and easily finding targets.

One regional surveillance method to ensure maximum coverage is the *lawnmower* path definition, sometimes referred to as a *boustrophedon* pattern. The pattern is efficient for ensuring maximum coverage of an area. However, it is very time consuming, and depending on the requirement of the mission, may be ineffective for the needs of the operator. Similarly, a spiral pattern that slowly spirals in either smaller or larger radius could provide similar results.

One challenge is determining how long or how many times a vehicle must survey a point before a satisfactory level of confidence that a target exists in a given area. Bertuccelli and How [76] proposed a robust UAS search method for determining target existence with the consideration that the prior probabilities for a given cell are poorly known. The use of Beta distribution enabled a prediction of the number of searches required in a given cell to achieve the desired confidence that a target exists in a given area.

Qu *et al.* [61] proposed a pheromone-based algorithm with an artificial potential field to perform regional surveillance with multiple UASs. A region would be separated into multiple units and a pheromone model would be applied to each unit. Pheromones have a diffusion feature that results in a portion of its information being translated to the units around it, using either an attractive or repulsive factor. This information then results in a gradient of pheromones being formed providing a path for the UAS to follow. An artificial potential field was then used to aid in obstacle avoidance, collision avoidance, and optimal search.

A planning algorithm by Song *et al.* [77] for optimal monitoring of spatial environmental phenomena based on Gaussian process priors showed improvement at finding global maximum conditions. This algorithm would be valuable for surveying unknown spatio-temporal fields such as gas plumes and humidity. Lee and Morrison [78] propose a search algorithm for multiple-vehicle maritime search and rescue that accounts for target drift using a mixed integer linear program, relying on a model over multiple periods to account for object location over time.

Zhang and Pei [79] developed a method to track the boundary of an oil spill using model predictive control and universal kriging. Universal kriging is an interpolation technique closely related to regression analysis. By combining universal kriging and model predictive control they proposed a method to search the environment with a sensor and, based upon the initial samplings, develop a means to track the boundary of the oil spill.

Hu *et al.* [80] provided a multi-agent information fusion and control scheme for target searching. An individual probability map for target location(s) was maintained by each vehicle and updated, based on measurements made by the vehicle, using Bayes' rule. A consensus-like distribution fusion scheme, updated with asynchronous information, was used to create a multi-agent probability map for target existence. A distributed multi-agent coverage control method for path planning, using a Voroni partition, that ensured a sufficient number of visits to each cell was performed.

Hirsch and Schroeder [81, 82] proposed a method of decentralized cooperative control of multiple-UASs performing multiple tasks in an urban environment. The construct assumed limited communication between the vehicles and considered potential line of sight impacts from buildings. The method required each UAS to perform independent receding horizon feedback control that relied on its own information along with any received remote information from neighbor vehicles to plan the required search path.

Multiple Objective

As a vehicles mission progresses, the need to adapt to new information or to change objectives may be required. Additionally, balancing multiple internal objectives such as path length, endurance, and safety present challenges to the algorithm development and usage for UAS control. The ability of a system to perform mission re-tasking and path re-planning is needed to enable multiple-objective scenario use. Numerous path planning algorithms discussed earlier integrate object avoidance algorithms. Other safety considerations (such as air collision avoidance) would require unique algorithms that are discussed separately.

Multiple-objective path planning deals with integrating multiple path-planning algorithms into one framework to enable a vehicle to perform multiple missions, either in a hierarchical fashion or simultaneously. Multiple-scenario response can be due to the need to respond to changing environments, as seen in Meng *et al.* [24]. They proposed a hierarchical approach that removed and replaced mission objectives as the mission requirements are changed or canceled. The approach developed an initial path generation for each vehicle and then, as requirements changed, re-allocation of objectives was performed with each UAS receiving a unique tasking and path generation.

Hirsch and Schroeder [81, 82] defined a solution for vehicles performing tasks of searching for targets while also tracking targets already found with a hybrid heuristic algorithm that combined a greedy randomized adaptive search procedure with simulated annealing (GRASP-SA). The UASs were provided no knowledge of where or how many targets were present in the environment. At each decision point, the UAS was required to determine whether to continue searching for new targets or track the targets already detected. The approach was performed in an urban environment model, incorporating object avoidance and line-of-sight obstructions into the decision process. The GRASP-SA algorithm was successfully applied to the problem set and provides a unique approach to the multiple-objective problem for search and tracking of multiple targets.

The use of Tabu search as a method for multiple-objective planning was proposed by Wang [21]. This early evaluation of the problem requires additional work but showed promise as a way to plan missions for multiple-task planning with goals of maximizing number of completed tasks with a minimization of range and time. The Tabu search algorithm is used to optimize the task allocation scheme after a planning model is built. The problem set evaluated was simple, and requires a more complex and practical environment to be evaluated in order to determine the overall value of this approach.

An algorithm utilizing multi-criteria decision making cost functions and multi-attribute utility theory to make complex decisions for vehicle path planning was designed by Wu *et al.* [83] with a focus on UAS delivery of medical supplies while flying in a complex airspace. The approach focused on flying a vehicle under existing visual flight rules in the national airspace, which continues to be a critical concern. For en-route planning, multiple criteria were considered within the cost construct of the algorithm: time, fuel, airspace classes, aircraft separation risk, storm cell risk, cruising levels, and population risk. These concerns, while more tactically focused on completing a singular mission type (medical delivery), could be transformed into other objectives in a similar construct for more complex efforts.

Ilaya [84] proposed the use of a decentralized control scheme involving multiple vehicles performing a multi-objective trajectory tracking and consensus problem using particle swarm optimization. The approach incorporated a two-level decision process: a high-level supervisory level and a local vehicle control level. Decentralized model predictive control was utilized for the vehicle-level synthesis of cooperative and self behaviors. A Lie group of flocks approach was used for the high-level supervisory control decision making. In related work, Ilaya *et al.* [85] provided an approach for distributed and cooperative decision making for collaborative electronic warfare. Similar algorithms were utilized with a focus on radar deception, ensemble tracking, and collision avoidance among the vehicles.

Optimizing resources for multi-criteria decision making using ant colony optimization (ACO) and analytic hierarchy process (AHP) was proposed by Fallahi *et al.* [59]. Unlike other approaches that rank a finite set of alternatives in a multi-criteria decision making problem, this approach utilizes the ACO to obtain optimal solutions satisfying some of the path-planning criteria. The AHP is then used to select the best UASs to perform each portion of the mission, optimizing the results of the overall mission. This approach could be extended to numerous UAS problems and objective constructs.

Peng *et al.* [86] proposed using a linkage and prediction dynamic multi-objective evolutionary algorithm in conjunction with a Bayesian network and fuzzy logic decision making process. Historical Pareto sets are collected and analyzed for the online path planning. A Bayesian network and fuzzy logic are then utilized for bias calculations for each objective. Results of using this method shows improved performance over completely restarting the path-planning algorithm at each objective change.

Multiple Aircraft

Multiple aircraft path planning deals with both centralized and decentralized coordinated mission efforts of multiple vehicles. Significant research is currently being performed in this area and could constitute a full survey paper on its own. The goal of this section is to highlight some of the primary methods being proposed for single and multiple task allocation that show significant capabilities of interest. Many of the previously discussed methods incorporated multiple vehicle controls into their algorithms, and are not repeated here. Specifically, the algorithms discussed in Section 2.4.1 also supported multiple-aircraft control.

Swarming is not considered in detail for this review as the controls for swarming have been extensively reviewed [87,88] and are generally focused on multiple vehicles working towards a single task while acting in a more biological-system manner. The term *swarm* is used in multiple ways currently to describe different operations. For this paper, a swarm is a group of vehicles working towards a common task in a group manner. This section focuses on multiple-UASs performing unique cooperative single and multiple-tasks in a controlled but decentralized environment.

One significant area of research has been at MIT under Professor Jonathan How. An unbiased Kalman consensus algorithm was proposed by Alighanbari and How [89]. Consensus-based algorithms proposed by How and associates consider both a decentralized consensus-based auction algorithm (CBAA) and the consensus-based bundle algorithm (CBBA). The CBAA is used for single-assignment tasking of single agents using an auction with greedy heuristics and a conflict-resolution protocol for consensus on winning bids for allocation [90]. The CBBA algorithm allows each agent to bundle assignments awarded as with CBAA but enables the system to collect and perform multiple assignments [91, 92]. Both algorithms show significant value in allocation of tasks and the overall performance of the system of UASs to complete missions in complex environments.

A survey of early consensus problems for multi-agent coordination and development of consensus seeking algorithms was completed by Ren and Beard [93–95]. Extensions of some of this work have included forest fire monitoring using multiple UASs [96] and perimeter surveillance using teams of UASs [97].

Zhao and Zhao [22] propose task clustering as a means to divide a large portion of tasks among multiple UASs. Ou *et al.* [98] propose a chaos optimization algorithm for task assignment to multiple-UASs. Zhang *et al.* [99] propose a cooperative and geometric learning path-planning algorithm for single and multiple UASs that attempts to minimize both the risk and length of the path flown by the vehicle(s).

2.4.2 SAFETY CONTROLS

Safe control of UASs is a critical area of concern. Manned aircraft have the unique advantage of having a pilot in the loop directly at the vehicle with the ability to react to safety concerns immediately. UASs require either automated response capabilities or the ability to quickly provide critical information to an operator for response. The notion of run-time assurance to provide confidence in autonomous decisions is a critical area of concern. Collision control is a huge concern for UASs, which has resulted in work on air, ground, and object avoidance. Boundary and population control work has focused on keeping UASs in controlled areas and out of the area of risk to populations. Weather avoidance enables a vehicle to evaluate the weather and determine whether the flight path should be modified autonomously. Fault tolerance, isolation, and prognostics is a large research area focused on enabling vehicles to continue to operate under less than ideal functionality. Flight termination is a major concern for how to manage vehicles that are not operating properly and must immediately cease flight operations. Test safety is a unique area of concern focused on how to properly test vehicles prior to release to normal operations where the likelihood of failure is higher and risks can be increased.

Run-Time Assurance

A critical safety concern for autonomous systems is trust in decision making. As the ability of autonomy to make decisions, the state space of the system grows so large that it is impossible to verify and validate all possible decisions made by an autonomous system. As a result, a way to ensure that the system does not make decisions outside an acceptable region is required. The concept of run-time assurance has been investigated by the Air Force Research Lab [100, 101] and is supported by research by Barron Associates [102]. Mark Skoog at NASA Armstrong has proposed an Expandable Variable-Autonomy Architecture (EVAA) system architecture that would enable confidence in UAS decision making by bounding the decision making to prevent a system from operating in unsafe manners [103]. Research continues to be performed in trust of autonomous systems, including in manned systems, such as autonomous ground collision avoidance systems on fighter aircraft [104–107]. Extending the confidence in autonomous unmanned systems will be critical in the future. Ensuring that the system architecture enables safety, including implementing a run-time assurance concept as in the EVAA architecture, will be critical to ensure safe operations.

Collision Avoidance

Collision control has been a key concern for all aircraft, but is more difficult on a UAS due to the lack of onboard pilots. The vehicle must be able to avoid collision with the ground, with other vehicles, and any objects it may encounter. Developing algorithms to perform vehicle maneuvering for collision avoidance can be performed by methods previously discussed in Section 2.4.1. The difficulty in collision avoidance is identification of the risk and determining what mitigation must be performed. The challenge is identifying what sensor or model is required to determine the risk and what mitigations can be implemented.

Ground collision avoidance and recovery systems were originally developed for manned aircraft. In 1990 a patent was granted for an ”Aircraft ground collision avoidance and autorecovery systems device" [108] which provided a system design for calculating aircraft flyups to prevent ground collision. Automatic Ground Collision research efforts [109–112] continued in the USAF and NASA for years. In early 2015, an Automatic Ground Collision Avoidance System installed on the F-16, which was based upon the earlier research by the USAF and NASA, saved the pilot and aircraft during operations against the Islamic State in Syria [113]. Recently, NASA [114] integrated a similar system on a small UAS with a smart-phone interface for ground collision avoidance capabilities. These systems rely upon an onboard digital terrain elevation model of the earth to calculate threat and recovery, nullifying the need for additional sensors on the vehicle. Avoiding colliding with stationary objects is an additional consideration in the ground collision avoidance area. Scherer *et al.* [115] proposed implementation of Laplace's equation to develop a potential field solution. Kobilarov [116] utilized a cross-entropy method in conjunction with rapidly expanding random trees for object avoidance path generation. Hrabar [117] proposed using an expanding elliptical search method to determine paths for object avoidance.

Air collision avoidance of both aircraft and stationary objects is a more difficult problem to solve. A patent was filed in 2001 [118] that provided the initial concept of a safety zone sphere around a vehicle that required both passive and active sensors to monitor the safety zone. Once incursion was detected, the system would inform an onboard sense-and-avoid computer for corrective action response. This underlying philosophy has been utilized in numerous research efforts. One of the critical issues for air collision avoidance is awareness of air traffic within the local airspace. For vehicles that do not have sufficient sensors or capabilities, a ground-based sense-and-avoid system (GBSAA) has been proposed for numerous aircraft types [119]. The Department of Defense, under an Army program, is in the process of developing and deploying several GBSAA systems in specific areas around the US [120]. However, there are challenges to using GBSAA for UAS operations, including the potential that they may not be able to actually support collision avoidance between vehicles [121]. The additional disadvantage of a GBSAA system is that it still requires integration with the UAS or have a pilot in control of the vehicle to make avoidance decisions. Maneuver algorithms are rather straightforward to implement depending on the path requirements; however, integration of GBSAA sensor data is more difficult. Most sensors installed on vehicles or the ground are only sensing other vehicles that may threaten the vehicle of concern. The integration of that information along with vehicle path planning and control for collision avoidance is required. One approach for being able to identify where all traffic is for collision avoidance sensing is the requirement by the FAA to have Automatic Dependent Surveillance-Broadcast (ADS-B) on all aircraft by the year 2020 [122]. Lin and Saripalli [123] proposed a method of collision avoidance utilizing ADS-B with a greedy rapidly exploring random tree.

Numerous methods have been proposed and explored as ways to prevent collisions between UASs and other aircraft. Considering collision avoidance an optimization problem and implementing geometry-based solutions has been proposed by multiple sources [124–126]. Lin and Saripalli [127] proposed using reachable sets, a collection of locations that can be reached at a given instant in time, for collision avoidance. Lin and Saripalli [128] also proposed using a variation of a rapidly expanding random trees approach with 3D Dubins Curves to avoid both stationary and moving targets. Optimal control methods were proposed by Shim and Sastry [129] and Bareiss and van den Berg [130] that require accurate system models to be implemented. Methods utilizing Markov decision processes were proposed in [131–133] but can be limiting due to the large state space requirements. Jackson *et al.* [134] propose a sensor suite of both onboard and offboard sensors with sensor fusion as a method to detect threats. Numerous methods of integrated sensor solutions exist for identifying air collision threats including Traffic-alert and Collision Avoidance System (TCAS) [135], mobile radar [136], electro-optical/infrared [137], and Laser & Light Detection and Ranging (LIDAR) [138]. Angelov [139] provides a significant review of the work being performed in this area. In 2009, the US Office of Naval Research performed a detailed study of sensor solutions for sense and avoid [140].

Boundary Control

Geofence is a general concept of providing boundary control of where a vehicle can operate. For UASs this can include ceilings and floors as well as walls of operation. Geofencing is seen as one means of aiding in the sense and avoid issue of UASs, especially small UASs.

Stevens *et al.* [141] proposed a geofence system that was platform independent. Many of the small UAS flight controls systems, such as Arduino [142], include a geofence capability inherent in its system that can be utilized for safe operations. Hayhurst *et al.* [143] propose that standards be developed for assured containment that provides an independent capability separate from the geofence algorithms that are generally resident internal to system software.

Test Safety

The Range Safety Group of the Range Commanders Council has provided guidelines for Flight Safety Systems for UAS Operations within defined range locations [144]. This standard provides guidelines for safe recovery of UASs that are recommendations for flying on these ranges. Johns Hopkins Advanced Physics Laboratory has proposed a framework for safe testing, Safe Testing of Autonomy in Complex, Interactive Environments (TACE), that can provide both safety features and complex interactive environments in a virtual fashion enabling improved safety and testability [145]. Emergency recovery and flight termination requirements are a necessary capability of UASs, especially larger class UASs that can cause danger to property or personnel. A technology survey of these systems by Stansbury *et al.* [146] provide a good overview of the recovery and termination systems. The need to terminate or provide emergency recovery is critical, especially when testing unproven systems. Integrating emergency recovery and/or flight termination capabilities with systems such as TACE for testing of vehicles will be a critical enabler to safety of the system and any property and personnel close to the test area. Including enabling features like geofencing/boundary control, collision avoidance techniques, reversionary modes to disable autonomy algorithms, and the ability to take over the vehicle during undesirable operations will be critical for safe testing of these systems. The size and level of implementation of recovery and termination systems along with systems like TACE and EVAA along with considerations of run time assurance will be dependent upon vehicle size and risk.

2.5 IMPROVEMENT AREAS

The use of UASs will continue to grow in the future. To be able to address growing and changing needs in the future, a system will need to be able to adapt and change to new and emerging requirements. A significant amount of work has been performed to develop algorithms that will enable path planning for multiple mission types. As discussed, the majority of this work has been performed with key assumptions about vehicle capability and simplified performance metrics. In order to improve the capabilities of these systems, the integration of system capabilities and performance must be included in the algorithms. Development of controls for UASs that include estimates of vehicle performance and capabilities will be key to improving the usefulness of the algorithms in actual systems. To date, limited evaluation of existing algorithms with multiple vehicle types and their associated dynamics has been performed. Additionally, integrating vehicle performance capabilities and limitations into the algorithms will result in a better understanding of the capabilities and value of the different algorithms in realistic implementations.

The ability to integrate multiple mission types in a single vehicle will be critical as development costs and time lines continue to challenge the market. Developing system designs and algorithms to account for multiple-scenario missions will enable the growth of UASs with reduced delay and costs. Development of a hierarchical scheme for both single and multiple UASs needs to continue in order to enable future systems. The integration of key system safety requirements, especially in the area of collision avoidance and boundary control, will help ensure that these vehicles are safe and trustworthy to operate. Developing overarching safety architectures that ensure safe operations will be critical to building trust in autonomy and enabling run-time assurance. Ground collision avoidance is becoming standard with new capabilities in operation, but air collision and object avoidance still has development work to be performed to ensure safe and trusted operations.

While not discussed in detail in this paper, sensors and sensor data fusion will be critical areas that need to be addressed to truly enable autonomous control methods. Until the UAS can self evaluate the data that it has received and make the appropriate decisions on that data, the usefulness of the vehicles will be mostly limited to man-in-the-loop operations. The size and capability of sensors, and the vehicles they will be integrated in, will be critical to the growth and development of UASs for future use.

2.6 CONCLUSIONS

The development of autonomous UASs and subsystems will continue to expand in the future. The systems being developed will need to be safe, reliable, and effective across multiple operational environments and tasks. Development of systems that can perform multiple scenarios and be adaptable to new capabilities and responsive to changes in environments and missions will be key to future success. Significant research and development in capabilities has been performed and continues. A modular system architecture was proposed in this paper that will enable safe and trustworthy performance of multiple-scenario missions. Critical path-planning and safety controls which will provide underlying system capabilities for future development were reviewed. Future integration of actual UAS performance capabilities, including costs for vehicle dynamics, into the proposed architecture and evaluation of numerous path-planning algorithms will enable a better understanding of existing methods and future capabilities on actual systems.

CHAPTER 3

ROBUST UAV PATH PLANNING USING POMDP WITH LIMITED FOV SENSOR

3.1 INTRODUCTION

The growth in use of unmanned systems has resulted in an explosion in control methods. Unmanned aerial vehicles (UAVs) provide the opportunity to perform many of the dull, dirty, and dangerous missions that are ill-suited for manned systems. In order to effectively and efficiently perform these missions, robust algorithms to control these UAVs must be developed. Significant work is being performed in numerous control areas for UAVs [147]. One method of path planning control that has shown significant promise is the use of decentralized partially observable Markov decision processes (POMDP) with a nominal belief-state optimization (NBO) approximation [148].

We design a UAV control method that utilizes POMDPs with NBO approximation utilizing a fixed field of view (FOV) sensor that has limited observation windows. We build upon our previous efforts [148–150], that have focused on the ability of the POMDP to provide sufficient path planning for a UAV tracking of ground based fixed or moving targets. During initial development, it was assumed that the UAV(s) could always see the targets, giving rise to a restrictive model for the partially observable portion of the POMDP. We incorporate a fixed FOV sensor to determine the performance of the POMDP with limited observations. We vary the fixed look angles and the altitude to determine the performance of the method while tracking one or two targets. The performance is evaluated using the tracking errors and percentage of observations across the configurations.

Previous work [148] has shown the ability of POMDP with NBO to perform effective path planning for tracking fixed and moving targets. Additionally, collision avoidance and wind compensation schemes were also shown to be effective using the same POMDP algorithms [149]. The POMDP with NBO approach uses a receding horizon method that is computationally efficient and provides effective responses to changes in target dynamics or environmental changes. The contribution of this paper is to show that the POMDP with NBO approach provides robust path planning and target tracking with *limited observations* due to the fixed FOV of the sensor. In this paper, we focus on non-evasive moving targets with limited observations.

3.2 PROBLEM SPECIFICATION

The ground targets move in 2-D. A simplified UAV motion model is used, utilizing forward acceleration and bank angle for control application with constant altitude. The UAVs have fixed FOV sensors mounted underneath the vehicle that provide limited observation measurement. These measurements are corrupted by spatially varying random errors, dependent upon UAV and target locations. The UAV has a limited speed range that can be varied by controlling the forward acceleration. The limited bank angle of the UAV can be adjusted to change the heading, but also impacts the sensor FOV. The UAV altitude can be set for a given scenario, but will maintain a constant altitude during that entire scenario. Changing the UAV altitude is effectively the same as changing the sensor size or FOV. The objective is to minimize the mean-squared error between the tracks and targets.

3.3 POMDP AND NBO APPROXIMATION

A POMDP is a discrete time controlled dynamical process that is useful for modeling resource control problems. A POMDP can be interpreted as a controlled version of a hidden Markov reward process. The NBO approximation is a method that has been shown to be computationally efficient for guidance optimization.

3.3.1 POMDP INGREDIENTS

States: The POMDP states represent time evolving system features. Three subsystems are defined: the sensors, the targets, and the tracker. At time k, the state is given by $x_k = (s_k, \chi_k, \xi_k, \mathbf{P}_k)$, where s_k is the sensor state, χ_k is the target state, and $(\xi_k, \mathbf{P}_k)$ is the state of the tracker. The sensor state provides the velocities and position of the UAV, and the target state provides the velocities, positions, and accelerations of the targets. The tracker state is a standard Kalman filter [151, 152], with ξ_k the posterior mean vector and $\mathbf{P}_k$ the posterior covariance matrix.

Actions: The actions are controls available to the system. The actions for this problem are the forward acceleration and the bank angle of the UAV. Specifically, the action at time k is given by $u_k = (a_k, \phi_k)$, with a_k and ϕ_k containing the forward acceleration and bank angle, respectively, for the UAV.

Observations and Observation Law: The states of a POMDP are not fully observable; only a random observation of the underlying state is available at any given time. Let χ_k^{pos} be the position vectors of a target and s_k^{pos} be the position vectors of a sensor/UAV. The observation of the target's position can be expressed by

$$z_k^{\chi} = \begin{cases} \chi_k^{pos} + w_k, & \text{if target is visible} \\ \text{no measurement}, & \text{otherwise} \end{cases} \tag{1}$$

where w_k is a random measurement error whose distribution is dependent on the UAV (s_k^{pos}) and the target (χ_k^{pos}) locations. Sensor and tracker states are assumed to be fully observable.

State-Transition Law: The state-transition law defines the next-state distribution given a current state and action pair. It is convenient to separately define the state-transition law for each of the three predefined subsystems. The sensor state progresses by $s_{k+1} = \psi(s_k, u_k)$, where ψ is a mapping function (defined later). The target state evolves according to $\chi_{k+1} = f(\chi_k) + \nu_k$,where ν_k is an independent and identically distributed (IID) random noise sequence and f represents the target motion model (also defined later). The tracker state evolves according to the Kalman filter equations with joint probabilistic data association (JPDA) [151, 153]. When target observations are not available, only the prediction step in the Kalman filter is performed, the update equation is not evaluated.

Cost Function: The cost function defines the cost of taking an action in a given state. Our cost function considers the mean-squared error between the tracks and the targets: $C(x_k, u_k) = E_{\nu_k, w_{k+1}}[\|\chi_{k+1} - \xi_{k+1}\|^2 | x_k, u_k]$

Belief State: The belief state is the posterior distribution of the underlying state, which is incrementally updated via Bayes rule given the observations. The belief state at time k is given by $b_k = (b_k^s, b_k^\chi, b_k^\xi, b_k^{\mathbf{P}})$, where $b_k^s = \delta(s - s_k)$, $b_k^\xi = \delta(\xi - \xi_k)$, $b_k^{\mathbf{P}} = \delta(\mathbf{P} - \mathbf{P}_k)$ (because the sensor and tracker states are fully observable), and b_k^χ is the target state posterior distribution.

3.3.2 OPTIMAL POLICY

The objective of the optimization policy, given the POMDP formulation, is to choose actions over a time horizon $H, k = 0, 1, ..., H - 1$, that minimize the expected cumulative cost. The expected cumulative cost over the time horizon H can be written as $J_H = E[\sum_{k=0}^{H-1} C(x_k, u_k)]$. The action chosen at time k should depend on the history of all observable quantities until time $k - 1$. If an optimal choice of actions exists, then there exists an optimal action sequence that depends only on "belief-state feedback" [154]. Therefore, the objective function can be written in terms of the belief states as follows: $J_H = E[\sum_{k=0}^{H-1} c(b_k, u_k) | b_0]$, where $c(b_k, u_k) = \int C(x, u_k) b_k(x) dx$.

According to Bellman's celebrated principle of optimality [155], the optimal objective function value J_H^* given the current belief state b_0 can be written as follows: $J_H^*(b_0) = \min_u \{c(b_0, u) + E[J_{H-1}^*(b_1) | b_0, u]\}$, where b_1 is the random next belief state, J_{H-1}^* is the optimal cumulative cost over the horizon $H - 1, k = 1, 2, ..., H - 1$, and $E[\cdot | b_0, u]$ is the conditional expectation given the current belief state b_0 and an action u taken at time $k = 0$. The Q-value of taking an action u given the current belief state b_0 is defined by $Q_H(b_0, u) = c(b_0, u) + E[J_{H-1}^*(b_1) | b_0, u]$. The optimal policy (from Bellman's principle) at time $k = 0$ can be written as $\pi_0^*(b_0) = \text{argmin}_u Q_H(b_0, u)$. The optimal policy at time k is $\pi_k^*(b_k) = \text{argmin}_u Q_{H-k}(b_k, u)$.

In practice, the second term in the Q-function is difficult to obtain exactly. Numerous methods have been studied [150, 156–162] to approximate the Q-values. We use the NBO approximation method, which was introduced in [150] along with other guidance problem approximations and techniques.

3.3.3 NBO APPROXIMATION

Although there are a number of approximation methods available to solve POMDPs, we chose the NBO method because it is less computationally expensive compared to other POMDP approximation methods like Q-learning, policy rollout, hindsight optimization, and foresight optimization [154]. In practice, a UAV guidance algorithm needs to be implementable in real-time, requiring a method that is not computationally prohibitive. In our previous work [148], we showed that the NBO algorithm provided an acceptable computation quality.

Assume there are N_{targs} targets. We represent the target state as $\chi_k = (\chi_k^1, \chi_k^2, ..., \chi_k^{N_{\text{targs}}})$, where χ_k^i represents the ith target. The track-state is $\xi_k = (\xi_k^1, \xi_k^2, ..., \xi_k^{N_{\text{targs}}})$ and $\mathbf{P}_k = (\mathbf{P}_k^1, ..., \mathbf{P}_k^{N_{\text{targs}}})$, where $(\xi_k^i, \mathbf{P}_k^i)$ is the track-state corresponding to the ith target. We use a linearized target motion model with zero-mean noise to model the target-state dynamics, as given below $(\forall i)$

$$\chi_{k+1}^i = \mathbf{F}_k \chi_k^i + \nu_k^i, \qquad \nu_k^i \sim N(0, \mathbf{Q}_k) \tag{2}$$

with the observations as follows: $z_k^{\chi^i} = \mathbf{H}_k \chi_k^i + w_k^i$ if the target is visible, and no measurement otherwise, where $w_k^i \sim N(0, \mathbf{R}_k(\chi_k^i, s_k))$, $\mathbf{F}_k$ is the target motion model (same for all targets), and $\mathbf{H}_k$ is the observation model (1) (same for every target) according to which only the position of a target is observed. The state of the ith target (χ_k^i) includes its 2-D position coordinates (x_k, y_k), its velocities (v_k^x, v_k^y) and accelerations (a_k^x, a_k^y) in x- and y-directions, i.e., $\chi_k^i = [x_k, y_k, v_k^x, v_k^y, a_k^x, a_k^y]^T$. Therefore, the observation model is $\mathbf{H}_k = [\mathbf{I}_{2x2}, \mathbf{0}_{4x4}]$. We adopt a constant velocity (CV) model [151, 152] for target dynamics in (2), which defines $\mathbf{F}_k$. The belief state corresponding to the ith target, based upon assumed Gaussian distributions, can be expressed as $b_k^{\chi^i}(\chi) = N(\chi - \xi_k^i, \mathbf{P}_k^i)$, where ξ_k^i and $\mathbf{P}_k^i$ are the track-states of the ith target, which evolve according to the JPDA algorithm [151, 153].

The NBO method approximates the objective function as $J_H(b_0) \approx \sum_{k=0}^{H-1} c(\hat{b}_k, u_k)$, where $\hat{b}_1, \hat{b}_2, ..., \hat{b}_{H-1}$ is a nominal belief-state sequence and the optimization is over the action sequence $u_0, u_1, ..., u_{H-1}$. The nominal belief-state sequence for the ith target can be identified with the nominal tracks $(\hat{\xi}_k^i, \hat{\mathbf{P}}_k^i)$, which are obtained from the Kalman filter equations [31, 32] with exactly zero-noise sequence as follows: $\hat{b}_k^{x^i}(\chi) = N(\chi - \hat{\xi}_k^i, \hat{\mathbf{P}}_k^i)$, $\hat{\xi}_{k+1}^i = \mathbf{F}_k \hat{\xi}_k^i$, and

$$\hat{\mathbf{P}}_{k+1}^i = \begin{cases} [[\hat{\mathbf{P}}_{k+1|k}^i]^{-1} + \mathbf{S}_{k+1}^i]^{-1} & \text{if measurement available} \\ \hat{\mathbf{P}}_{k+1|k}^i & \text{otherwise} \end{cases} \tag{3}$$

where $\hat{\mathbf{P}}_{k+1|k}^i = \mathbf{F}_k \hat{\mathbf{P}}_k^i \mathbf{F}_k^T + \mathbf{Q}_k, \mathbf{S}_{k+1}^i = \mathbf{H}_{k+1}^T [\mathbf{R}_{k+1}(\hat{\xi}_{k+1}^i, s_{k+1})]^{-1} \mathbf{H}_{k+1}$ and $s_{k+1} = \psi(s_k, u_k)$ (ψ is defined in the next subsection). In (3), the nominal error covariance matrix $\hat{\mathbf{P}}_{k+1}^i$ is dependent upon the availability of future observations. Because the availability of these observations is uncertain, we can guess by assuming the location of the target at time $k+1$ as $\hat{\xi}_{k+1}^{i,pos}$ (component of nominal track-state corresponding to the ith target at time $k+1$) and checking its line of sight from the sensor location, i.e., s_{k+1}^{pos}. The cost function, i.e., the mean-squared error between the tracks and the targets, can be written as $c(\hat{b}_k, u_k) = \sum_{i=1}^{N_{\text{targs}}} \text{Tr}\hat{\mathbf{P}}_{k+1}^i$. The goal is to find an action sequence $(u_0, u_1, ..., u_{H-1})$ that minimizes the cumulative cost function (truncated horizon [150]) $J_H(b_0) = \sum_{k=0}^{H-1} \sum_{i=1}^{N_{\text{targs}}} \text{Tr}\hat{\mathbf{P}}_{k+1}^i$, where $\hat{\mathbf{P}}_{k+1}^i$ represents the nominal error covariance matrix of the ith target at time $k+1$. Here, we adopt a "receding horizon control" approach, where we optimize the action sequence for **H** time-steps from the current time-step but implement only the action corresponding to the current time-step followed by an optimization of the action sequence for **H** time-steps in the next time-step.

Our approach is related to model predictive control (MPC), as argued by the authors of [163]. According to [163], the MPC method is a type of rollout algorithm (an approximation method to solve Markov decision processes (MDPs) and POMDPs) with a particular base policy, where the stability property of MPC is a special case of the cost improvement property of rollout algorithms that employ a sequentially improving base policy. In other words, MPC can also be viewed as an approach to solve a POMDP.

The measurement error, i.e., w_k in (1), has a normal distribution $N(0, \mathbf{R}_k(\chi_k, s_k))$, where $\mathbf{R}_k$ reflects $p\%$ range uncertainty and q-rad angular uncertainty. If r_k is the distance between the target and the sensor at time k, then the standard deviations corresponding to the range ($\sigma_{\text{range}}(k)$) and the angle ($\sigma_{\text{angle}}(k)$) are $\sigma_{\text{range}}(k) = (p/100)r_k$ and $\sigma_{\text{angle}}(k) = qr_k$. The information matrix depends on the inverse of the measurement covariance matrix, which depends on the distance between the sensor and the target. Therefore, the information matrix blows up when the UAV is exactly on top of the target (i.e., when $r_k = 0$ the sensor's location overlaps with the target's location in our 2-D environment). To address this problem, we define the effective distance (r_{eff}) between the sensor and the target as follows: $r_{\text{eff}}(k) = \sqrt{r_k^2 + b^2}$, where r_k is the actual distance between the target and the sensor and b is some non-zero real value. Therefore, the standard deviations of the range and the angle are given by $\sigma_{\text{range}}(k) = (p/100)r_{\text{eff}}(k)$ and $\sigma_{\text{angle}}(k) = qr_{\text{eff}}(k)$. If θ_k is the angle between the target and the sensor at time k, then $\mathbf{R}_k$ is calculated as follows:

$$\mathbf{R}_k = M_k \begin{bmatrix} \sigma^2_{\text{range}}(k) & 0 \\ 0 & \sigma^2_{\text{angle}}(k) \end{bmatrix} M_k^T \qquad \text{where}$$

$$M_k = \begin{bmatrix} \cos(\theta_k) & -\sin(\theta_k) \\ \sin(\theta_k) & \cos(\theta_k) \end{bmatrix}.$$

The eigenvalues of the matrix $\mathbf{R}_k$ are therefore $\sigma^2_{\text{range}}(k)$ and $\sigma^2_{\text{angle}}(k)$.

3.4 UAV KINEMATICS AND FIXED FOV

Simplified UAV kinematics are used for calculating the UAV motion. With utilization of a fixed FOV sensor, the FOV can be calculated in camera axes. Using Euler angle transformations, it can be determined if the target is within the sensor FOV.

3.4.1 UAV KINEMATICS

In this subsection we define the mapping function ψ introduced in Section III to describe the evolution of the sensor (UAV) state given an action, i.e., $s_{k+1} = \psi(s_k, u_k)$. The state of the ith UAV at time k is given by $s_k^i = (p_k^i, q_k^i, V_k^i, \theta_k^i)$, where (p_k^i, q_k^i) represents the position coordinates, V_k^i represents the speed, and θ_k^i represents the heading angle. Let a_k^i be the forward acceleration (control variable) and ϕ_k^i be the bank angle (control variable) of the UAV, i.e., $u_k^i = (a_k^i, \phi_k^i)$. The mapping function ψ can be specified as a collection of simple kinematic equations that govern the UAV motion. The kinematic equations of the UAV motion are as follows. The speed is updated according to $V_{k+1}^i = [V_k^i + a_k^i T]_{V_{\text{min}}}^{V_{\text{max}}}$, where $[v]_{V_{\text{min}}}^{V_{\text{max}}} = \max\{V_{\text{min}}, \min(V_{\text{max}}, v)\}$, where V_{min} and V_{max} are the minimum and the maximum limits on the speed of the UAVs. The heading angle is updated according to $\theta_{k+1}^i = \theta_k^i + (gT\tan(\phi_k^i)/V_k^i)$, where g is the acceleration due to gravity and T is the length of the time-step. The position coordinate are updated according to $p_{k+1}^i = p_k^i + V_k^i T\cos(\theta_k^i)$ and $q_{k+1}^i = q_k^i + V_k^i T\sin(\theta_k^i)$.

3.4.2 FIXED FOV CALCULATION

Given a sensor width (x_{sens}) and height (y_{sens}), along with the focal length (f) of an installed sensor, a field of view can be calculated for a given altitude (alt). The angular field of view for the sensor width is $\text{FOV}_w = 2\tan^{-1}(x_{\text{sens}}/2f)$ and for the sensor height is $\text{FOV}_h = 2\tan^{-1}(y_{\text{sens}}/2f)$. Given a height above the ground, the edges of the field of view can be calculated in camera axes. It is assumed that the camera is installed concurrent to the body axis, for ease of calculation. The top and bottom edges of the FOV are $x_{FOV} = \pm(\text{alt})\tan(0.5\text{FOV}_h)$ and the left and right edges of the FOV are $y_{FOV} = \pm(\text{alt})\tan(0.5\text{FOV}_w)$, where alt is the altitude of the sensor.

In order to determine if an observation is made, the position of the target must be translated and rotated into the UAV body axis. A vector η is calculated between the UAV position and the target position, $\eta = (p_{\text{UAV}} - p_{\text{target}}, q_{\text{UAV}} - q_{\text{target}}, \text{alt}_{\text{UAV}} - \text{alt}_{\text{target}})$. Given the UAV Euler angles, pitch (θ), roll (ψ), and heading (ϕ) a rotation of η can be made between the world axis and body axis. The value of η in the body axis can now be compared to the FOV for the sensor. If the position

Table 3.1: Sensor Field of View

Altitude (m)	FOV Height (m)	FOV Width (m)
200	113	188
500	283	471
750	424	707
1000	566	943
1500	849	1415
2000	1132	1886

of the target, η, is within the field of view, an observation is made and the POMDP will calculate normally with an updated observation, as specified in (1). However, if η is not in the FOV, there will be no observation and no measurement will be passed to the POMDP.

3.5 TARGET TRACKING RESULTS

Monte Carlo runs were performed on several configurations to evaluate the effectiveness of our method to track the target. The two primary conditions were whether the UAV was tracking one or two targets. Three sensor angle configurations were evaluated: downward, 15 degrees forward, and 15 degrees right. Six altitudes were evaluated for each of the six configurations: 200, 500, 750, 1000, 1500, and 2000 meters. A range uncertainty of 10 percent and an angular uncertainty of 1 percent were used for this evaluation.

The sensor simulated was modeled after several commercially available sensors [164]. The sensor was assumed to have a focal plane width of 5 mm, height of 3 mm, and a lens focal length of 5.3 mm. This configuration provides a field of view of 50.5 degree wide and 31.6 degrees tall. Varying the altitude provides a different size of field of view. Table 3.1 provides the total height and width of the sensor field of view at each of the altitudes, based upon a straight lookdown angle. As the angle of the sensor is rotated from straight down, the actual ground plane field of view will change. It was assumed that all targets could be tracked at all altitudes.

The targets moved at a constant speed of 10 m/s at a constant heading. The UAV was allowed to vary speed between 16 and 25 m/s and a maximum bank angle of 30 degrees. In order to provide a consistent analysis baseline, altitude was only considered for variation of sensor field of view; otherwise the problem was treated as a 2-D problem. Tracking errors were not considered as a function of altitude, only of x-y distance variations to enable FOV comparison. Previous efforts have shown that the average location error was 2-3 meters for similar scenarios [149].

Overall, the algorithm showed a robust ability to track moving targets even with a limited FOV that decreased the observations and increased the tracking error. However, for general tracking of vehicles, the method shows the ability to adequately track single and multiple targets for observation purposes. The algorithm also showed computational efficiency. The POMDP calculation for a random run averaged 0.46 seconds on an Intel Core I7-6700, 4 GHz, 64-bit processor with 24 GB of Ram.

3.5.1 SINGLE TARGET TRACKING

Single target tracking with limited field of view sensors showed effective tracking of a single moving target. Figure 3.1 shows an example run of a UAV tracking the moving target. The figure shows when the target was observed (green circles), when it was not observed (red x) and what the predicted location of the vehicle was by the POMDP algorithm (black dot). The line with asterisks shows the general flight path of the UAV with the location recorded at 2 second intervals.

The quality of the tracking can normally be related to the quantity of observations. The greater the observation percentage, the lower the location tracking error will be. This trend can be seen in Figure 3.2 for downward and side facing sensor configurations. However, it can be noted that the location error increased with increasing observations for the forward facing sensor configuration.

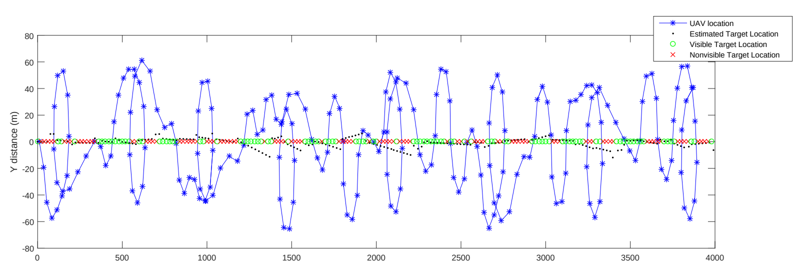

Figure 3.1: Tracking Example: 1 Target, 1 Sensor, 200 m, 15 deg fwd sensor

The forward facing sensor does not show a decreasing error due to the geometric conditions that result in observations. The decreased tracking accuracy, even under high quality observations, can be related to the geometry of the observations. The observations may be made but the change in observation may not significantly improve the tracking error, resulting in larger errors than other configurations.

In general, the increase in altitude results in an increase in the number of observations and a reduction in the location error. This characteristic relationship is expected, because as the altitude increases, the size of the observation window increases, as shown in Table 3.1. However, as can be seen, even with a relatively large decrease in the percent of observations seen at lower altitudes, there is not a significant increase in the tracking error. As expected, the POMDP algorithm was able to provide sufficient estimates of target location during non-observation periods such that there was not a dramatic increase in tracking errors.

Overall, for single target tracking, the downward sensor shows the best overall performance, followed by the right looking sensor. This result is what is expected when you consider that the downward and right facing sensor configurations provide more observations at lower altitudes and provide better geometry for observation updates. Ultimately, the goal of having quality observations for the moving target is met for all configurations. If having the lowest error is a primary consideration, then the downward sensor configuration is likely the best choice for single target tracking.

3.5.2 TWO TARGET TRACKING

Tracking two targets is well suited for the POMDP algorithm. The increased potential of missed observations due to the fixed FOV and separated targets can be addressed by the POMDP predictions. Figure 3.3 provides an example of a nominal two target tracking run with a 15 degree forward sensor angle. The plot is derived similarly to Figure 3.1, except there are two targets being tracked, as indicated by two separate lines, as opposed to one.

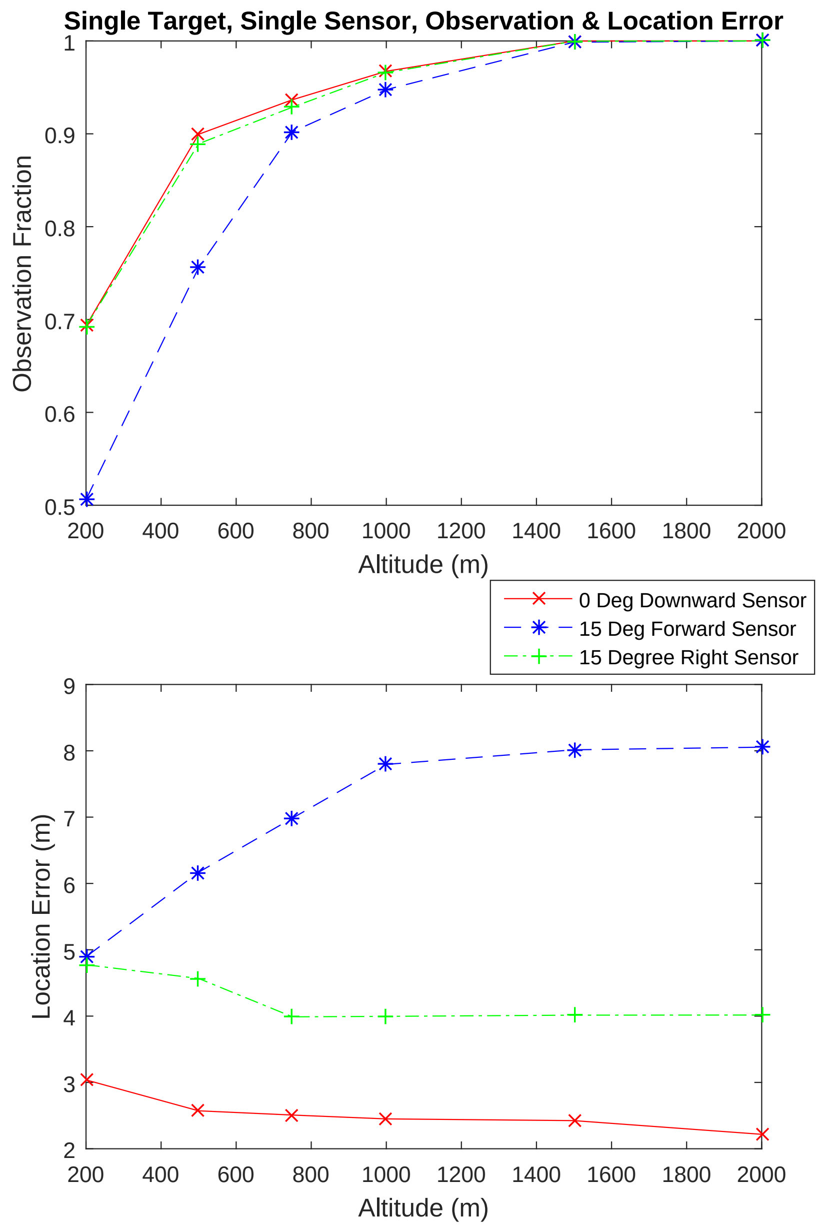

Figure 3.2: Average Location Error and Observation Fraction: 1 Target, 1 Sensor, 1000 Runs

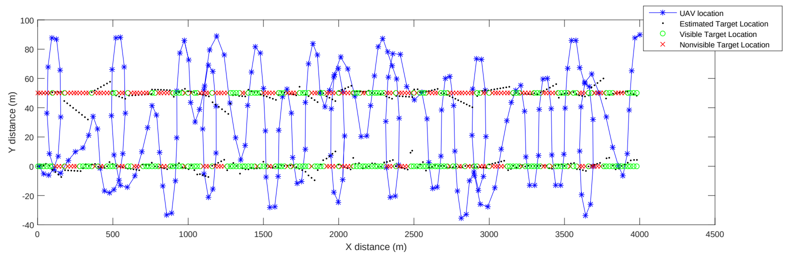

Figure 3.3: Tracking example: 2 Targets, 1 Sensor, 200 m, 15 deg fwd sensor

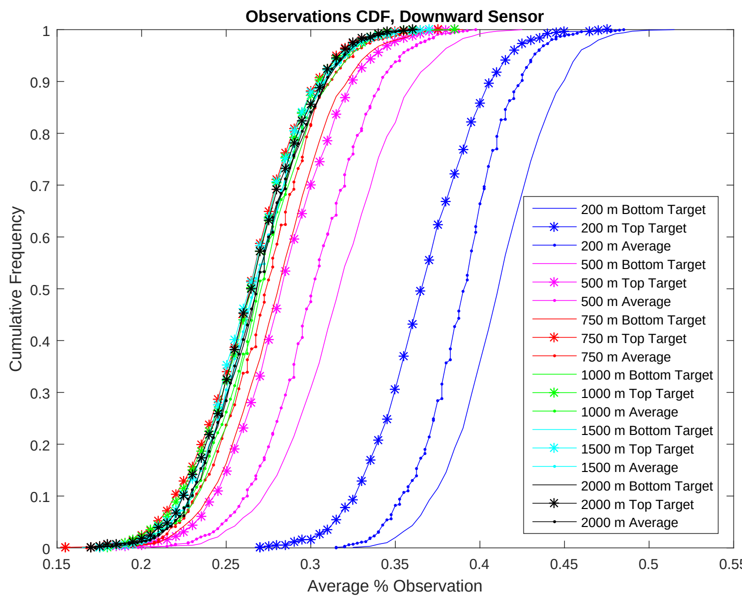

Figure 3.4: Tracking performance at various altitudes for each target

Because two targets are being tracked, the errors and observations can be presented separately or averaged. For comparisons in this paper, the average errors for each target were averaged together instead of reported separately. Similarly, the percentage of observations were averaged. Figure 3.4 shows the cumulative distribution function (CDF) for each altitude, for a downward looking sensor. A CDF provides the representation of how often (frequency) within the data set the average observation was at or less than the observation percent at the point. As can be seen, the CDF for each target is well behaved, as are the averages of the two targets at a given altitude. This behavior supports using the averages of the percent observation and the cumulative target error of both targets.

As was the case in single target tracking, the ability of the POMDP to track two targets shows promise. For all three sensor configurations, as shown in Figure 3.5, the increase in altitude results in a decrease in average tracking error. This trend is consistent with single target tracking and correlates with the expectation that increased altitude should provide increased observations. However, as can be see in Figure 3.5, the average percent observation actually decreases as the altitude increases for the downward looking sensor. The algorithm, in this case, attempts to cross over each target to reduce the tracking error for the target, then transverses to the other target. Crossing over and passing, along with the turns, would result in less time in the sensor FOV, especially for a directly downward looking sensor

Overall, for two target tracking, the downward sensor shows the best overall performance, followed by the forward looking sensor. This result is what is expected when you consider that the downward and forward facing sensor configurations provide more simultaneous observations for the two target configuration. If having the lowest error is a primary consideration, then the downward sensor configuration is likely the best choice for single target tracking.

3.6 CONCLUSIONS

Overall, the performance of the POMDP algorithm is similar to previous results. The sensor model used here, specifically the use of a limited FOV sensor, is a more realistic use case for small UAV applications that do not have gimbaled sensor capabilities. The ability to track both single or dual moving targets can be performed well with different sensor configurations. The POMDP can address the increased number of missed observations under certain circumstances and still provide adequate predictions of the target location(s) until observations can be made again. While some sensor configurations may be preferred for certain tracking cases, all configurations showed acceptable performance to be able to maintain a track on the target(s).

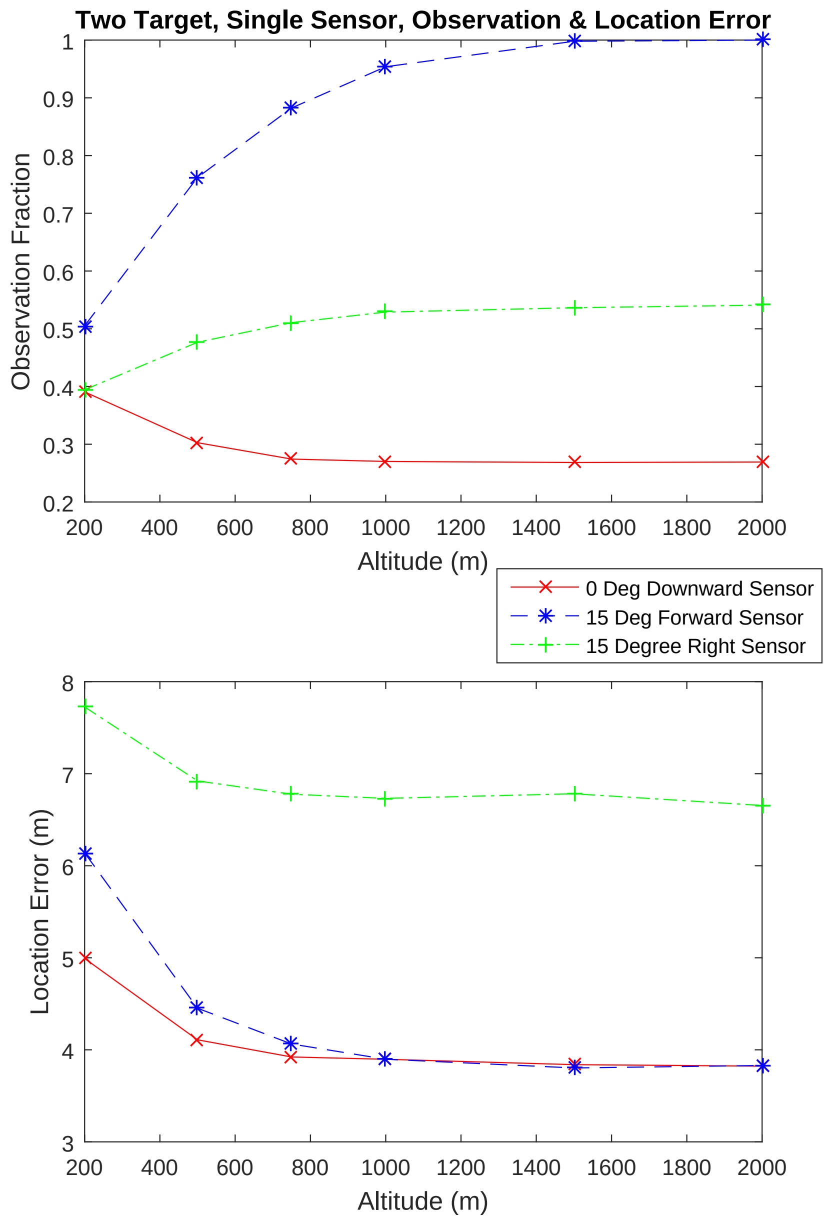

Figure 3.5: Average Location Error and Observation Fraction: 2 Target, 1 Sensor, 1000 Runs

CHAPTER 4

FUEL EFFICIENT MOVING TARGET TRACKING USING POMDP WITH LIMITED FOV SENSOR

The ability to effectively track moving targets is a critical capability for future autonomous aircraft. While many methods have been developed for performing target tracking, minimal work has focused on fuel-efficient options to extend mission duration. The ability to tightly track a target is critical for certain missions; however, increased tracking errors can be accepted in certain scenarios to extend endurance. Partially Observable Markov Decision Processes (POMDPs) have been shown to be effective for tracking fixed and moving targets. This paper provides a fuel-efficient option that shows a 10% endurance increase with adequate target tracking. The algorithm provides tracking with a limited field of view fixed sensor that will have limited observations depending on mission requirements. The POMDP formulation proposed in this paper is robust enough to handle observations while also providing options for improved fuel efficiency. We[1] perform 500 Monte Carlo simulations per configuration to provide statistical confidence in the performance of the algorithm .

4.1 INTRODUCTION

Unmanned aerial vehicles (UAVs) are well suited for performing dull, dirty, and dangerous missions normally performed by manned aircraft. One mission of critical interest is tracking moving targets. Significant research has been performed in numerous UAV control areas including path planning for target tracking [147]. However, the majority of path planning algorithms developed to date are focused on completing a specific mission task with a focus on tracking accuracy,

[1]The research in this chapter was done in collaboration with Lucas Krakow [167]

target coverage, or expediency of mission completion. This approach provides satisfactory performance but generally result in less than desired UAV dynamics and limited endurance. There is limited consideration for fuel efficiency of the vehicle in the decision process of most of the algorithms. Ignoring the fuel efficiency concern can result in shorter missions or limit the mission effectiveness.

Previously, we had presented a robust path planning algorithm for a limited field of view sensor utilizing a partially observable Markov decision process (POMDP) with a nominal belief-state optimization (NBO) approximation [165], [148], [166]. These efforts have shown the ability of POMDP with NBO to perform effective path planning for tracking fixed and moving targets with a limited field of view (FOV) sensor. Additionally, collision avoidance and wind compensation schemes were also shown to be effective using the same POMDP algorithms [149]. The POMDP with NBO approach uses a receding horizon method that is computationally efficient and provides effective responses to changes in target dynamics or environmental changes. This method also provided a robust capability for tracking the targets across a range of altitudes and sensor configurations.

Here, we design a UAV control method that uses POMDPs with NBO approximation for a fixed FOV sensor with fuel efficiency considerations. We build upon our previous efforts [148–150,165], that have shown the POMDP provides sufficient path planning for a UAV tracking of ground based fixed or moving targets. The cost function is modified to add in variable fuel burn considerations to improve mission endurance. The performance is evaluated using the tracking errors and the fuel burned during the simulations for multiple fuel burn options. The contribution of this paper is to show that the POMDP with NBO can provide effective target tracking while improving fuel efficiency. In this paper, we focus on non-evasive moving targets with limited observations.

4.2 PROBLEM SPECIFICATION

Ground targets moving in 2-D at constant speed will be tracked. A simplified UAV motion model with forward acceleration and bank angle controls is used. The UAV has a limited speed range that is controlled by commanding longitudinal acceleration. The UAV bank angle is used to change UAV heading, while also changing the sensor observation area. A fixed FOV sensor mounted on the bottom of the vehicle provides a limited observation measurement of target location. Spatially varying random errors, based on UAV and target locations, corrupt the accuracy of the target location measurements. The fuel burn of the UAV is calculated based on the speed and bank angle of the UAV. The UAV altitude can be set for a given scenario, but will maintain a constant altitude during that entire scenario. Changing the UAV altitude is effectively the same as changing the sensor size or FOV. The objective is to provide a spectrum of options to balance the minimization of the mean-squared tracking error with the fuel efficiency of the UAV.

4.3 POMDP AND NBO APPROXIMATION

A POMDP is a discrete time controlled dynamical process that is useful for modeling resource control problems. A POMDP, unlike a Markov Decision Process that has perfect knowledge of the system state, has some state information that is not directly observed. The NBO approximation method has been shown as computationally efficient for guidance optimization. The method described here builds on our earlier papers [148–150, 165, 166].

4.3.1 POMDP INGREDIENTS

States: The POMDP states represent time evolving system features. Three subsystems are defined: the sensors, the targets, and the tracker. At time k, the state is defined as $x_k = (s_k, \chi_k, \xi_k, \mathbf{P}_k)$, where s_k is the state of the sensor, χ_k is the target state, and $(\xi_k, \mathbf{P}_k)$ is the tracker state. The sensor state is the UAV position and velocities, and the target state is the target position, velocities, and accelerations. The tracker state is a posterior mean vector, ξ_k, and posterior covariance matrix, $\mathbf{P}_k$, of a standard Kalman filter [151, 152].

Actions: The actions for this problem, or the available system controls, are UAV forward acceleration and the bank angle. Specifically, at time k the forward acceleration, a_k, and bank angle, ϕ_k, define actions $u_k = (a_k, \phi_k)$ of the UAV.

Observations and Observation Law: The target states are not fully observable; at any given time a random observation of the target state may be available. Let χ_k^{pos} be the target position vector and s_k^{pos} be the sensor/UAV position vector. The observation of the target's position can be expressed by:

$$z_k^{\chi} = \begin{cases} \chi_k^{pos} + w_k, & \text{if target is visible} \\ \text{no measurement}, & \text{otherwise} \end{cases} \tag{1}$$

where the random measurement error, w_k, has a distribution dependent on target and UAV locations. Full observability is assumed for Sensor/UAV and tracker states.

State-Transition Law: The state-transition law defines the next-state distribution of the three predefined subsystems given a current state and action pair of each subsystem. The sensor state evolves by $s_{k+1} = \Psi(s_k, u_k)$, where Ψ is a mapping function (defined later). The target state progresses according to $\chi_{k+1} = f(\chi_k) + \nu_k$, with independent and identically distributed random noise sequence, ν_k, and a target motion model, f (also defined later). Kalman filter equations with joint probabilistic data association (JPDA) provide tracker state evolution [151, 153]. The update equation is only evaluated when target observations are available, otherwise only the prediction step in the Kalman filter is performed.

Cost Function: Our cost function, which is the action cost for a given state, considers the mean-squared error between the tracks and the targets: $C(x_k, u_k) = E_{\nu_k, w_{k+1}}[\|\chi_{k+1} - \xi_{k+1}\|^2 | x_k, u_k]$

Belief State: The underlying state posterior distribution, updated incrementally via Bayes rule with observations, defines the belief state. At time k, the belief state is $b_k = (b_k^s, b_k^{\chi}, b_k^{\xi}, b_k^{\mathbf{P}})$, where $b_k^s = \delta(s - s_k)$, $b_k^{\xi} = \delta(\xi - \xi_k)$, $b_k^{\mathbf{P}} = \delta(\mathbf{P} - \mathbf{P}_k)$ (because of full tracker and sensor state observability), and b_k^{χ} is the posterior distribution of the target state.

4.3.2 OPTIMAL POLICY

The objective is to choose actions that minimize the expected cumulative cost over a time horizon $H, k = 0, 1, ..., H - 1$. For a time horizon H, the expected cumulative cost can be written as $J_H = E[\sum_{k=0}^{H-1} C(x_k, u_k)]$. Historical knowledge of all observable quantities up to time $k-1$ should inform the chosen action at time k. If an optimal choice of actions exists, then there exists an optimal action sequence that depends only on "belief-state feedback" [154]. Therefore, the belief states can be used to write the objective function as: $J_H = E[\sum_{k=0}^{H-1} c(b_k, u_k)|b_0]$, where $c(b_k, u_k) = \int C(x, u_k) b_k(x) dx$.

Bellman's principle of optimality [155] provides for the optimal objective function value J_H^* given the current belief state b_0. The optimal objective function can be written as: $J_H^*(b_0) = \min_u\{c(b_0, u) + E[J_{H-1}^*(b_1)|b_0, u]\}$, where b_1 is the random next belief state, J_{H-1}^* is the optimal cumulative cost over the horizon $H-1, k = 1, 2, ..., H-1$, and $E[\cdot|b_0, u]$ is the conditional expectation given the current belief state b_0 and an action u taken at time $k = 0$. Given the current belief state b_0, the Q-value of taking an action u is defined by $Q_H(b_0, u) = c(b_0, u) + E[J_{H-1}^*(b_1)|b_0, u]$. At time $k = 0$ the optimal policy, from Bellman's principle, can be written as $\pi_0^*(b_0) = \text{argmin}_u Q_H(b_0, u)$. The optimal policy at time k is $\pi_k^*(b_k) = \text{argmin}_u Q_{H-k}(b_k, u)$.

The second Q-function term is difficult to obtain exactly. Studies of numerous methods to approximate the Q-values have been performed [150, 156–162]. The NBO approximation method, introduced in [150] along with other guidance problem approximations and techniques, is used here.

4.3.3 NBO APPROXIMATION

Although a few approximation methods to solve POMDPs are available, we chose the NBO method because of low computationally cost relative to other POMDP approximation methods like foresight optimization, hindsight optimization, policy rollout, and Q-learning [154]. In practice, real-time implementation of a UAV guidance algorithm requires a method that is not computationally prohibitive. Our previous work [148, 165] showed acceptable computation efficiency for the NBO algorithm.

Given N_{targs} targets, the target state is represented as $\chi_k = (\chi_k^1, \chi_k^2, ..., \chi_k^{N_{\text{targs}}})$, where the ith target is represented by χ_k^i. The track-state is $\xi_k = (\xi_k^1, \xi_k^2, ..., \xi_k^{N_{\text{targs}}})$ and $\mathbf{P}_k = (\mathbf{P}_k^1, ..., \mathbf{P}_k^{N_{\text{targs}}})$, where $(\xi_k^i, \mathbf{P}_k^i)$ is the ith target track-state. We use a zero-mean noise linearized target motion model for the target-state dynamics, given by $(\forall i)$

$$\chi_{k+1}^i = \mathbf{F}_k \chi_k^i + \nu_k^i, \qquad \nu_k^i \sim N(0, \mathbf{Q}_k) \tag{2}$$

with observations as defined in (1). The measurement error is $w_k^i \sim N(0, \mathbf{R}_k(\chi_k^i, s_k))$, and the target motion model (for all targets) is $\mathbf{F}_k$. The ith target state (χ_k^i) includes position coordinates (x_k, y_k), velocities (v_k^x, v_k^y), and accelerations (a_k^x, a_k^y) in x- and y-directions, i.e., $\chi_k^i = [x_k, y_k, v_k^x, v_k^y, a_k^x, a_k^y]^T$. The observation model becomes $\mathbf{H}_k = [\mathbf{I}_{2x2}, \mathbf{0}_{4x4}]$. A constant velocity (CV) model [151, 152] is implemented for target dynamics in (2), which defines $\mathbf{F}_k$. The ith target belief state, with assumed Gaussian distributions, can be expressed as $b_k^{\chi^i}(\chi) = N(\chi - \xi_k^i, \mathbf{P}_k^i)$, where ξ_k^i and $\mathbf{P}_k^i$ are the ith target track-states, which evolve according to the JPDA algorithm [151, 153].

The objective function is approximated by the NBO method as $J_H(b_0) \approx \sum_{k=0}^{H-1} c(\hat{b}_k, u_k)$, where $\hat{b}_1, \hat{b}_2, ..., \hat{b}_{H-1}$ is a nominal belief-state sequence over an optimized action sequence $u_0, u_1, ..., u_{H-1}$. The ith target nominal belief-state sequence, developed from nominal tracks $(\hat{\xi}_k^i, \hat{\mathbf{P}}_k^i)$ using a zero-noise Kalman filter [151, 152] is: $\hat{b}_k^{x^i}(\chi) = N(\chi - \hat{\xi}_k^i, \hat{\mathbf{P}}_k^i)$, where $\hat{\xi}_{k+1}^i = \mathbf{F}_k \hat{\xi}_k^i$, and

$$\hat{\mathbf{P}}_{k+1}^i = \begin{cases} [[\hat{\mathbf{P}}_{k+1|k}^i]^{-1} + \mathbf{S}_{k+1}^i]^{-1} & \text{if measurement available} \\ \hat{\mathbf{P}}_{k+1|k}^i & \text{otherwise} \end{cases} \tag{3}$$

where $\hat{\mathbf{P}}_{k+1|k}^i = \mathbf{F}_k \hat{\mathbf{P}}_k^i \mathbf{F}_k^T + \mathbf{Q}_k$, $\mathbf{S}_{k+1}^i = \mathbf{H}_{k+1}^T [\mathbf{R}_{k+1}(\hat{\xi}_{k+1}^i, s_{k+1})]^{-1} \mathbf{H}_{k+1}$ and $s_{k+1} = \Psi(s_k, u_k)$ (Ψ is defined in the next subsection). The nominal error covariance matrix $\hat{\mathbf{P}}_{k+1}^i$ in (3) is dependent upon the availability of future observations. Because of the uncertainty of future observations we can check for target observability by guessing the target location at time $k+1$ as $\hat{\xi}_{k+1}^{i,pos}$ and checking its line of sight from the sensor location (s_{k+1}^{pos}), where $\hat{\xi}_{k+1}^{i,pos}$ is the ith target nominal track-state at time $k+1$. We can write the cost function, defined as the mean-squared error between the targets and the tracks, as: $c(\hat{b}_k, u_k) = \sum_{i=1}^{N_{\text{targs}}} \mathrm{Tr}\hat{\mathbf{P}}_{k+1}^i$. We want to find the sequence of actions

$(u_0, u_1, ..., u_{H-1})$ that minimizes the cumulative cost function (truncated horizon [150]) $J_H(b_0) = \sum_{k=0}^{H-1} \sum_{i=1}^{N_{\text{targs}}} \text{Tr}\hat{\mathbf{P}}_{k+1}^{i}$, where $\hat{\mathbf{P}}_{k+1}^{i}$ represents the ith targets nominal error covariance matrix at time $k+1$. A "receding horizon control" approach is adopted, which optimizes the action sequence for **H** time-steps from the current time-step but only the action corresponding to the current time-step is implemented. At the next time-step, we perform a new action sequence optimization for **H** time-steps.

The measurement error, w_k in (1), is normally distributed $N(0, \mathbf{R}_k(\chi_k, s_k))$, where $\mathbf{R}_k$ contains q-rad angular uncertainty and $p\%$ range uncertainty. If the distance between the sensor and the target at time k is r_k, then the standard deviations corresponding to the range ($\sigma_{\text{range}}(k)$) and the angle ($\sigma_{\text{angle}}(k)$) are $\sigma_{\text{range}}(k) = (p/100)r_k$ and $\sigma_{\text{angle}}(k) = qr_k$. The information matrix depends on the inverse of the measurement covariance matrix, which depends on the distance between the sensor and the target. Therefore, the information matrix blows up when the UAV is exactly on top of the target (i.e., when $r_k = 0$ the sensor's location overlaps with the target's location in our 2-D environment). To address this problem, we define the effective distance (r_{eff}) between the sensor and the target as follows: $r_{\text{eff}}(k) = \sqrt{r_k^2 + b^2}$, where r_k is the actual distance between the target and the sensor and b is some non-zero real value. Therefore, the standard deviations of the range and the angle are given by $\sigma_{\text{range}}(k) = (p/100)r_{\text{eff}}(k)$ and $\sigma_{\text{angle}}(k) = qr_{\text{eff}}(k)$. If α_k is the angle between the target and the sensor at time k, then $\mathbf{R}_k$ is calculated as follows:

$$\mathbf{R}_k = M_k \begin{bmatrix} \sigma_{\text{range}}^2(k) & 0 \\ 0 & \sigma_{\text{angle}}^2(k) \end{bmatrix} M_k^T \qquad \text{where}$$

$$M_k = \begin{bmatrix} \cos(\alpha_k) & -\sin(\alpha_k) \\ \sin(\alpha_k) & \cos(\alpha_k) \end{bmatrix}.$$

The eigenvalues of the matrix $\mathbf{R}_k$ are therefore $\sigma_{\text{range}}^2(k)$ and $\sigma_{\text{angle}}^2(k)$.

4.4 UAV KINEMATICS AND FIXED FOV

Simplified UAV kinematics are used for calculating the UAV motion. Using a fixed FOV sensor, the FOV can be calculated in camera axes. Using Euler angle transformations, we can determined if the target is within the sensor FOV.

4.4.1 UAV KINEMATICS

In this subsection we define the mapping function Ψ introduced in Section III to describe the evolution of the sensor (UAV) state given an action, i.e., $s_{k+1} = \Psi(s_k, u_k)$. The state of the ith UAV at time k is given by $s_k^i = (p_k^i, q_k^i, V_k^i, \theta_k^i)$, where (p_k^i, q_k^i) represents the position coordinates, V_k^i represents the speed, and θ_k^i represents the heading angle. Let a_k^i be the forward acceleration (control variable) and ϕ_k^i be the bank angle (control variable) of the UAV, i.e., $u_k^i = (a_k^i, \phi_k^i)$. The mapping function Ψ can be specified as a collection of simple kinematic equations that govern the UAV motion. The kinematic equations of the UAV motion are as follows. The speed is updated according to $V_{k+1}^i = [V_k^i + a_k^i T]_{V_{\text{min}}}^{V_{\text{max}}}$, where $[v]_{V_{\text{min}}}^{V_{\text{max}}} = \max\{V_{\text{min}}, \min(V_{\text{max}}, v)\}$, where V_{min} and V_{max} are the minimum and the maximum limits on the speed of the UAVs. The heading angle is updated according to $\theta_{k+1}^i = \theta_k^i + (gT\tan(\phi_k^i)/V_k^i)$, where g is the acceleration due to gravity and T is the length of the time-step. The position coordinate are updated according to $p_{k+1}^i = p_k^i + V_k^i T\cos(\theta_k^i)$ and $q_{k+1}^i = q_k^i + V_k^i T\sin(\theta_k^i)$.

4.4.2 FIXED FOV CALCULATION

Given a sensor width (x_{sens}) and height (y_{sens}), along with the focal length (f) of an installed sensor, a FOV can be calculated for a given altitude (z). The angular FOV for the sensor width is $\text{FOV}_w = 2\tan^{-1}(x_{\text{sens}}/2f)$ and for the sensor height is $\text{FOV}_h = 2\tan^{-1}(y_{\text{sens}}/2f)$. Given a height above the ground, the edges of the FOV can be calculated in camera axes. We assume that the camera is installed concurrent to the body axis, for ease of calculation. The top and bottom edges of the FOV are $x_{FOV} = \pm\text{z}\ \tan(0.5\text{FOV}_h)$ and the left and right edges of the FOV are $y_{FOV} = \pm\text{z}\ \tan(0.5\text{FOV}_w)$, where z is the altitude of the sensor.

To determine if an observation is made, the position of the target must be translated and rotated into the UAV body axis. A vector η is calculated between the UAV position and the target position as $\eta = (p_{\text{UAV}} - p_{\text{target}}, q_{\text{UAV}} - q_{\text{target}}, \text{z}_{\text{UAV}} - \text{z}_{\text{target}})$. Given the UAV Euler angles, pitch (θ_{UAV}), roll (ψ_{UAV}), and heading (ϕ_{UAV}) a rotation of η can be made between the world axis and body axis. The value of η in the body axis can now be compared to the FOV for the sensor. If the position of the target, η, is within the field of view, an observation is made and the tracker will update normally with an updated observation, as specified in (1). However, if η is not in the FOV, there will be no observation and no measurement will be passed to the tracker.

4.5 FUEL BURN COST FUNCTION

The nominal cost function often used for vehicle control is focused purely on the estimated target location error, based on the trace of the covariance P_k. This simple cost function results in significant dynamics of the UAV to try to precisely track the target for as accurate of a target location estimate as possible. However, as a result, the increased dynamics results in less endurance. There are times during missions that maintaining observations of the target is important, but the accuracy of the estimated location is less of a concern. In these cases, increasing the mission endurance of the UAV by decreasing the dynamics becomes a viable option.

Depending on the level of desired tracking accuracy versus the increased mission endurance, the cost function can be updated to include a fuel burn estimate. A fuel-burn **B** can be included in the cost function, with a set of ratio variables, λ_1 and λ_2, to vary the impact of the error compared to the fuel burn by:

$$J_H(b_0) = \sum_{k=0}^{H-1} \Big(\sum_{i=1}^{N_{\text{targs}}} \lambda_1 \text{Tr} \hat{\mathbf{P}}_{k+1}^{i} \Big) + \lambda_2 \mathbf{B} \tag{4}$$

The trace of the estimated covariance is calculated for all targets across the horizon of concern and the fuel burn is calculated by estimating the power usage of the UAV during the horizon.

4.6 WEIGHTED TRACE PENALTY

When using a limited FOV sensor there are conditions when the UAV may not be able to observe the target for an extended period of time. As a result, the POMDP algorithm may be unable to determine an adequate path plan, due to extended non-observation across the limited horizon. This can be exacerbated when trying to limit maneuvering for fuel efficiency. To determine a sufficient path we incorporate a *Weighted Trace Penalty (WTP)* term into the calculation of the next best UAV command. The WTP term was previously developed as an estimated cost to go (ECTG) term to deal with occlusions preventing observations [150]. We can estimate this growth with a WTP term, which is a product of the current covariance trace and the minimum distance to observability (MDO) for the non-observed target. The terminal cost or ECTG term using the WTP takes the form:

$$\hat{J}(b) = J_{\text{WTP}}(b) := \gamma D(s, \xi^{\boldsymbol{q}}) \text{Tr} P^{\mathbf{q}} \tag{5}$$

where γ is a positive constant, $\boldsymbol{q}$ is the index of the worst non-observed target. The MDO, $D(s, \xi)$, is the distance from the UAV to the closest point where the target becomes observable. The addition of the WTP results in a final cost function of:

$$J_H(b_0) = \sum_{k=0}^{H-1} (\sum_{i=1}^{N_{\text{targs}}} \lambda_1 \text{Tr} \hat{\mathbf{P}}_{k+1}^{i}) + \lambda_2 \mathbf{B} + \gamma D(s, \xi^{\boldsymbol{q}}) \text{Tr} \mathbf{P}^{q} \tag{6}$$

4.7 EXPERIMENTS

Statistical analysis, based upon 500 Monte Carlo simulations per configuration, was performed to evaluate the effectiveness of our method to track the target with varying fuel efficiency. The number of targets tracked, the altitude of the UAV, and the burn ratio of the cost function were all varied for the experiment. A single UAV tracked either one or two targets. Previous research with a fixed FOV sensor configuration has shown that above 1000 meters the impact of the sensor FOV is minimal. Therefore, only three altitudes were evaluated: 200, 500, and 1000 meters. A burn ratio (tracking accuracy:fuel burn, $\lambda_1 : \lambda_2$) of 1:0, 1:1, 1:5, 1:10, and 1:15 were used. A range uncertainty of 10 percent and an angular uncertainty of 1 percent were used for this evaluation. The sensor was fixed in a 90 degree straight down FOV configuration, relative to the UAVs nose. A horizon window of 6 steps, at 2 seconds per step, was used for the trajectory calculation. The trajectory calculation was performed every 4 seconds. A 400 second duration was used for each Monte Carlo simulation.

The sensor simulated was modeled after several commercially available sensors [164]. The sensor was assumed to have a focal plane width of 5 mm, height of 3 mm, and a lens focal length of 5.3 mm. This configuration provides a field of view of 50.5 degree wide and 31.6 degrees tall. It is assumed at the altitudes tested, the sensor is capable of resolving the target sufficiently for tracking.

The targets moved at a constant speed of 10 m/s at a constant heading (unknown to the tracker). The UAV was allowed to vary speed between 16 and 25 m/s and a maximum bank angle of 30 degrees. To provide a consistent analysis baseline, altitude was only considered for variation of sensor field of view; otherwise the problem was treated as a 2-D problem. Tracking errors were not considered as a function of altitude, only of x-y distance variations to enable FOV comparison. Previous efforts have shown that the average location error was 2-3 meters for single target and 5-6 meters for multiple target tracking in similar scenarios [165].

A basic fuel-burn calculation based upon actual power usage of a small UAV was used [168]. Measurements of real-time energy usage (in watt-hr) from a fixed-speed commercially available small UAV were taken while the vehicle was flying constant altitude and either wings level or in a full bank. Data showed that the power usage between wings level and full bank were essentially linear. For simplicity, it is assumed that the aircraft is symmetric such that left or right bank angle require the same energy. The burn was estimated across the speed range to extend from just a fixed speed aircraft. Table 4.1 provides simple lookup values for interpolating fuel burn.

Table 4.1: Fuel Burn Values

Velocity (m/s)	Wings Level (Watt-hr)	30 deg Bank (Watt-hr)
14	95	110
20	105	120
25	112	135
30	119	145
35	130	160

4.7.1 RESULTS

Overall, the algorithm showed reasonable ability to track moving targets while providing fuel-efficient options. As previously seen, at lower altitudes, the tracking errors are generally higher due to reduced observations. The modified cost function provided the ability to increase endurance by 10% while still providing tracking ability. At the higher altitudes, the tracking errors were not significantly impacted.

Based on the fuel usage model of the UAV, if the vehicle were to fly straight and level (most efficient possible) at a speed of 14 m/s for a 400 second window, the UAV would use 10.56 W, or 95 W for an hour. At 35 m/s, that would result in 14.4 W for 400 seconds, or 130 W for an hour. However, because the vehicle must maneuver, an increase in power usage is experienced.

The average tracking errors and fuel burn were calculated for each altitude and burn ratio configuration. Figure 4.1 provides the comparison of mean tracking error and mean fuel burn for a 400 second simulation for a single target. Figure 4.2 shows the mean percent of observations for each configuration compared to the mean location error. The symbols indicate which burn ratio was used, and the line connected to the given symbol indicate the altitude for the simulation. The trace:burn ratio provides the values of $\lambda_1 : \lambda_2$. As can be seen, as the fuel-burn ratio is increased, the fuel consumption is reduced while the tracking error is increased. At 200 meters the errors increase and observations reduce significantly compared to the higher altitudes. This is due to the limited FOV of the sensor reducing the number of observations. As can be seen at all three altitudes, the fuel-burn ratio reduces the overall fuel use while showing an increase in tracking

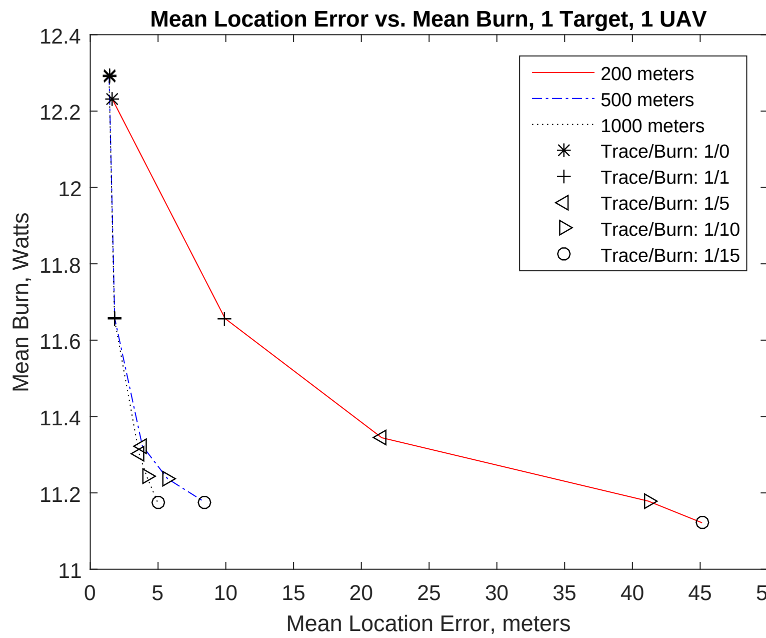

Figure 4.1: Average Location Error vs Fuel Burn, 1 UAV, 1 Target

errors, as expected. At the higher altitudes, the increase in tracking error is not as significant for the increased fuel efficiency. At 200 meters, however, a significant reduction in observations indicates increased lost tracks which drives a larger errors.

Figure 4.3 shows the mean tracking error versus mean fuel burn for two target tracking. Figure 4.4 shows the mean percent observations versus mean tracking error. The mean location errors and mean observation percentages are the average of all 500 runs for both targets. Only 500 and 1000 meter configurations are shown as the 200 meter data showed inadequate performance of tracking the second target. Over half of the runs at 200 meters indicated the UAV never made an observation of the second target. As can be seen, an increase in tracking error, corresponding to reduced observations, can be seen as we increase the fuel burn. At 1000 meters it can be noted that the change in fuel burn has a minimal impact on mean tracking errors.

While the 500 and 1000 meter tracking showed good performance, the 200 meter errors with one target became excessive and with two targets it was inadequate. The 200 meter tracking of a single UAV could be improved as well. One source of the poor performance is due to the MATLAB fmincon optimization function finding a local minimum and not the global minimum. Additional

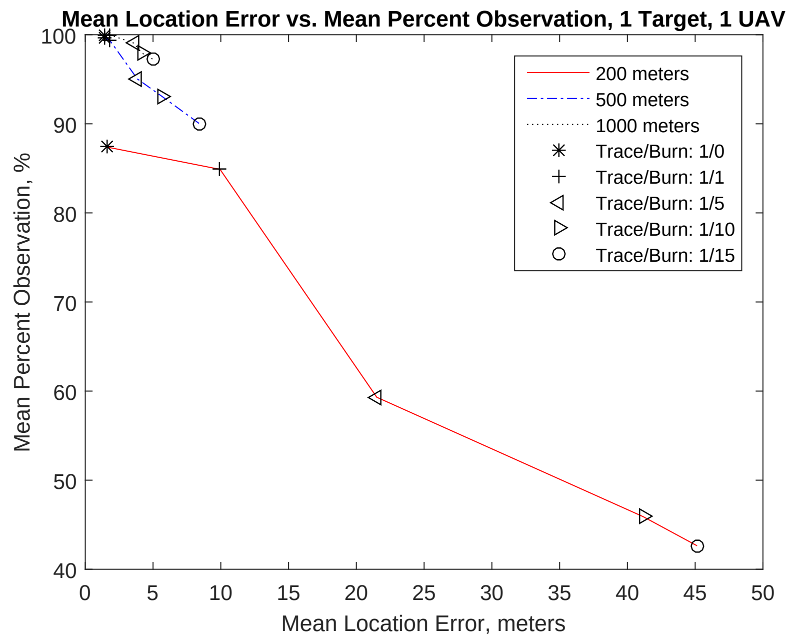

Figure 4.2: Average Location Error vs. Percent Observation, 1 UAV, 1 Target

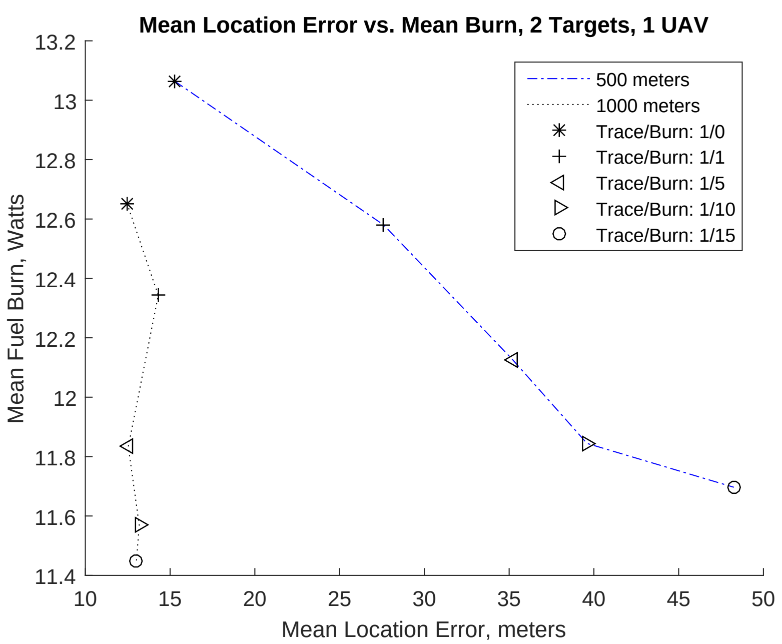

Figure 4.3: Average Location Error vs Fuel Burn, 1 UAV, 2 Targets

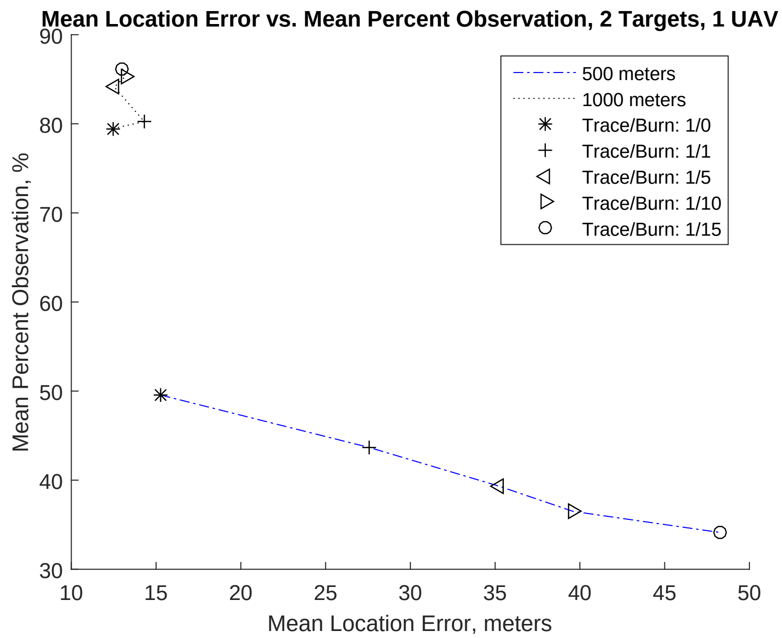

Figure 4.4: Average Location Error vs. Percent Observation, 1 UAV, 2 Targets

Table 4.2: Mean Fuel Burn

	$Single\ Target\ burn(W)$			$Two\ Targets\ burn(W)$	
$Ratio$	200 m	500 m	1000 m	500 m	1000 m
1 : 0	12.23	12.30	12.29	13.06	12.65
1 : 15	11.12	11.18	11.17	11.70	11.45
$\%gain$	9.98%	10.02%	10.03%	11.62%	10.48%

minimization options will be investigated for future development. Finally, due to the limited FOV of the sensor, the best way to improve the performance would be to either incorporate a sensor with a larger FOV or incorporate a gimbaled control of the sensor.

As discussed earlier, the fuel burn flying a pure straight and level 400 second flight would be 10.56 W. Table 4.2 shows the burns for single and two target tracking for each burn configuration. As can be seen, a 10+% increase in fuel efficiency is noted from the 1:0 configuration as compared to the 1:15 configuration.

4.8 CONCLUSIONS

The fuel efficient POMDP presented provides a robust tracking solution that enables fuel efficient options. Highly accurate tracking with a limited FOV sensor flying at low altitude is a difficult use case for UAVs. Significant maneuvering is required for a UAV to provide high quality tracking with minimal errors but leads to reduced endurance. Using a fuel-burn component in the cost function of the POMDP, alongside a Weighted Trace Penalty provides a highly capable tracker with improved endurance opportunities. Endurance increases of over 10% were experienced while still providing useful tracks of moving targets. Improvements in lower-altitude tracking may be seen by incorporating a gimbaled sensor and an improved optimization function.

CHAPTER 5

SERVICES BASED TESTING OF AUTONOMY

The test and evaluation (T&E) of autonomous systems, that adequately supports the verification and validation (V&V) process, is a significant challenge facing the test community. The ability to quickly and reliably test autonomy is necessary to provide a consistent T&E, V&V (TEVV) capability. A safe, efficient, and cost effective test capability, regardless of autonomy or sensor capability, is required. Autonomy and sensor capabilities, referred to as services, can be integrated easily into small Unmanned Aircraft Systems (sUAS) of differing capabilities and complexities. An integrated open-source architecture, for both software and hardware, implemented on multiple sUAS of varying capabilities can provide a robust test capability for emerging autonomous behaviors. The inclusion of a run time assurance (RTA) common safety watchdog and a Live-Virtual-Constructive (LVC) capability provides a consistent, robust, and safe test capability/environment. The use of an open software and hardware architecture ensures cross-platform viability. These features will allow test teams to focus on the newly incorporated autonomy and sensor services, not on other ancillary capabilities and systems on the test vehicle. Testing of the services in this manner will enable a common TEVV approach, regardless of final platform integration while decreasing risk and accelerating the availability of autonomy services. Services-Based Testing of Autonomy (SBTA) provides a cost-effective and focused capability to test autonomous services, whether software, hardware, or both.

5.1 INTRODUCTION

In April 2011, Secretary of Defense Robert Gates designated autonomy as one of seven technology priorities. In response, the United States Department of Defense (DoD) created an Autonomy Community of Interest (COI) to focus on the advancement of technology and capabilities to enable future autonomous systems [169]. In November 2014, Secretary of Defense Chuck Hagel [170] announced the Defense Innovation Initiative, which would help the DoD develop autonomy as part of a third offset strategy. Specifically, Secretary Hagel discussed areas of new technology research and development:

"Our technology effort will establish a new Long-Range Research and Development Planning Program that will help identify, develop, and field breakthroughs in the most cutting-edge technologies and systems âĂŞ especially from the fields of robotics, autonomous systems, miniaturization, big data, and advanced manufacturing, including 3D printing. This program will look toward the next decade and beyond."

Hagel's speech set the tone for future budget focus and research and development areas for the DoD. In June 2016, the Defense Science Board (DSB) Summer Study of Autonomy [171] concluded that autonomy had reached a "tipping point" in value.

One of the focus areas of the Autonomy COI was TEVV. The TEVV central technical challenge identified was: "From algorithms to scalable teams of multiple agents âĂŞ Developing new T&E, V&V technologies needed to enable the fielding of assured autonomous systems" [169]. In June 2015, the Autonomy COI TEVV working group released a Technology Investment Strategy 2015âĂŞ2018, which provided the framework for development of future TEVV needs and capabilities. This strategy provided five primary goals to align research of autonomy around:

1. Methods & Tools Assisting in Requirements Development and Analysis
2. Evidence-Based Design and Implementation
3. Cumulative Evidence through Research, Development, Test and Evaluation (RDT&E), Developmental Testing (DT), & Operational Testing (OT)
4. Run-Time Behavior Prediction and Recovery
5. Assurance Arguments for Autonomous Systems

The DSB autonomy study also noted that current DT and OT methods are in direct conflict with more advanced commercial methods that are better suited for testing of adaptive software [171].

The ability to bring in software very early to get user feedback via incremental upgrades interspersed with test is a key commercial capability. The DSB report recommended, among other things, that a new T&E paradigm for testing software that learns and adapts be established. In 2014, the National Research Council released a report on "Autonomy Research for Civil Aviation: Toward a New Era of Flight" that identified the need to develop standards and procedures for verification, validation, and certification of autonomous systems [172]. In 2016, the American Institute of Aeronautics & Astronautics Intelligent Systems Technical Committee developed a "Roadmap for Intelligent Systems in Aerospace" that identified the need for critical research in V&V methods to enable operations of adaptive systems in future vehicles [173]. To achieve improvement in the TEVV of autonomous systems, approaches to enable flexible and focused testing of autonomy are critical. The most challenging of components in these systems is the software embedded within them, specifically intelligent, learning, and adaptive software. While significant modeling and simulation, along with hardware in the loop testing, can be used, the ability to test autonomous capabilities in realistic scenarios is critical to the success of these types of systems. The cost and difficulty of performing significant flight testing of autonomous unmanned aircraft systems (UAS) are highly dependent on the vehicles and the integrated autonomous services. This paper provides the SBTA approach to testing these critical autonomy software and hardware capabilities, or autonomous services, in a rapid, repeatable, safe, and effective manner.

5.2 SBTA

Software and hardware to enable autonomy are critical focus areas for development of autonomous systems. A significant amount of research and development has been performed in the areas of path planning and safety controls algorithms [147]. Hardware, especially sensors to enable decision making, continue to be developed and improved, especially in the commercial arena. Autonomy services can be a combination of hardware and software to enable autonomous actions for a vehicle.

In most traditional acquisition programs, a new aircraft is procured with a significant amount of development of both the vehicle and the software. The advent of autonomy currently appears to be focused on developing capabilities to integrate into existing vehicles and systems. The cost to integrate autonomy into existing platforms can be expensive and time consuming. Providing a means to perform early testing of autonomy services is, therefore, critical. The intersection between the cyber and autonomy worlds raises yet another source of TEVV difficulty: can the perception sensors feeding the autonomy engine be spoofed (GPS, signature imagery, etc.) and can the autonomy engine understand that the "perceived world view" has been hacked?

Complex cyber-physical systems, such as autonomous UASs, are difficult to test with traditional methods currently used for standard UASs and manned aircraft. Testing of autonomy requires a different approach from initial software development through the end of life of the system. As autonomy becomes more complex, it becomes difficult to test all functions and capabilities due to the increasing size and complexity of the algorithms. Preventing the systems under test from getting into an undesirable or uncontrollable state while also challenging the autonomy is difficult. Providing a robust, reliable, and reusable approach to testing autonomy is key; this is why the SBTA method has been developed. This approach will provide a heterogeneous fleet of UASs that are simple and cost effective to modify and operate. These vehicles will be modified to have an open system architecture that will enable easy reconfiguration and installation of autonomy or other services that will enable testing. The approach is adaptable to different vehicles across a large operational envelope.

5.3 PLATFORM & OPERATIONS REQUIREMENTS

In April 2016, the United States Air Force (USAF) released the small Unmanned Aircraft Systems (sUAS) Flight Plan, which provided vision and strategy for development, operation, and sustainment of sUAS [174]. The USAF grouped sUAS into three categories, as defined in Table 1. These groups provide different levels of capability that are useful for not only actual operations but also for methods supporting TEVV.

Table 5.1: sUAS Platform Characteristics by Group

Group	Max Weigt (lbs)	Normal Ops Alt (ft AGL)	Speed (kts)
1	0-20	<1200	<100
2	21-55	<3500	<250
3	<1320	<FL 180	<250

The sUAS Flight Plan identifies significant future USAF operations for sUAS and autonomous capabilities. The ability to provide support across multiple scenarios to support numerous dull, dirty, and dangerous missions for the USAF is becoming more common. Additionally, the concept of interoperability and modularity between systems is a key capability for enabling future sUAS operations. Providing an open architecture capability for sUAS will enable these concepts. The affordability of acquisition and operations of sUAS is important not only for mission operations, but also for test and evaluation. Providing a test capability that meets these requirements will help enable future TEVV of autonomous services.

The commercial off the shelf (COTS) availability of numerous sUAS allows for rapid acquisition and cost effective operations. The majority of these COTS systems use existing commercially available controllers and autopilots that are easy to integrate and modify. Additionally, the ability to easily modify and add capabilities to these vehicles allows for flexible test assets. This modifiability and flexibility is a critical capability to enable SBTA. The only disadvantage to the majority of these systems is the limitation in size, weight, and power capabilities. The smaller the vehicle, the larger the limitation, primarily in the ability to add the autonomy which will require the integration of new and different sensors.

5.4 OPEN SOFTWARE & HARDWARE ARCHITECTURE

A service-oriented architecture (SOA) is a software design style that enables service provision to other components through a communication protocol [175]. The advantage of a SOA is that it provides a common interface for any developer to use. Providing a SOA-like interface for sUAS to enable integration of new autonomy services is a key to enabling SBTA. One SOA in development

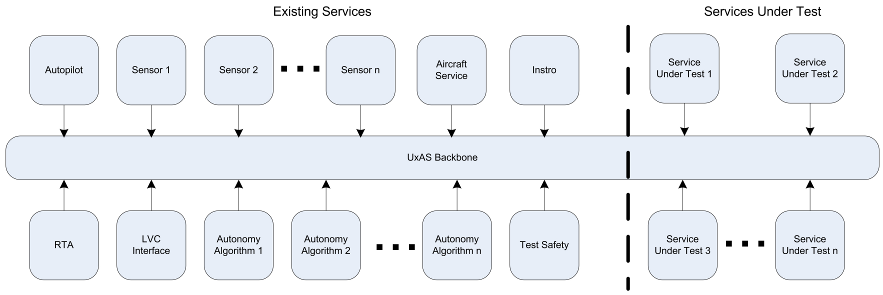

Figure 5.1: SBTA UxAS Architecture

is the Unmanned Systems Autonomy Services (UxAS) architecture being developed by the Air Force Research Laboratory (AFRL). The UxAS provides a common software interface architecture that can be integrated on any aircraft and provides some early autonomy capability developed by AFRL [176]. Recently, AFRL held a Summer of Innovation (SoI) event to improve the capabilities and documentation of UxAS [178] [179].

The majority of sUAS on the market use one of several standard autopilots for vehicle control. The UxAS architecture either already integrates with these standard autopilots or can easily be adapted via a new service interface. The AFRL has already generated multiple autonomy services on the UxAS architecture baseline that can be used for some mission scenarios. The results of the SoI provided improved capabilities and documentation of UxAS. Adopting a common architecture from initial development through flight testing on sUAS provides a robust TEVV capability.

Integrating the UxAS software onto a processor, such as a Raspberry Pi, oDroid, or Gumstix, and interfacing it with the sUAS autopilot will provide an initial capability for testing autonomy. Figure 5.1 provides an example UxAS architecture for SBTA. This architecture shows the segregation of existing known services and services under test. Having a UAS with an UxAS backbone and fully tested services will provide the framework for testing new services. These existing services will have known capabilities enabling testers to focus on the integration and performance of the new services under test.

5.5 RUN TIME ASSURANCE (RTA)

Assurance, as defined by the IEEE Standard Glossary of Software Engineering Terminology, is âĂIJa planned and systematic pattern of all actions necessary to provide adequate confidence that an item or product conforms to established technical requirementsâĂİ [180]. For most systems, assurance will be primarily completed during the design, development, and lab-testing phases of a project. Limited assurance testing is performed during flight testing, owing to the risks of putting a vehicle in an undesired condition. For autonomous systems with runtime variation in performance and decision, assurance that the system will not become inconsistent and unsafe is of critical importance [185].

To provide a safe and robust test asset, an independent watchdog supervisory monitor and controller to provide runtime assurance (RTA) coverage will be critical. In October 2017, ASTM International released ASTM F3269-17, Standard Practice for Methods to Safely Bound Flight Behavior of Unmanned Aircraft Systems Containing Complex Functions, which provides guidance for RTA architecture evidence on highly automated UAS [181]. While the primary concept provided in the specification relates to production RTA, the same concept can be exploited for test safety RTA. The ability to have both onboard and off-board RTA coverage will be critical during the testing of autonomous systems. These RTA watchdogs will need to provide user-defined capabilities to include loss of link, geo-fence, aircraft limit control, control override, object avoidance, and aircraft collision avoidance. Some of these systems may require off-board information (e.g., object and aircraft collision avoidance) while others will be required to be onboard (e.g., aircraft limits and geo-fence). Additionally, the ability to completely isolate the autonomy under test and revert to basic aircraft controls will be a critical, failsafe feature.

Several test aircraft have had similar capabilities installed on them for safety of test purposes. One example is the F-16 Variable stability In-flight Simulator Test Aircraft (VISTA) operated by the USAF Test Pilot School, which has also been used for early testing of autonomy [182]. This system allows changes to the commands, stability, and performance of the aircraft that can be turned off based upon pilot input or violation of any implemented safety limits. Similarly, Calspan has created a variable stability Learjet with a similar implementation [183]. Additionally, NASA Armstrong has been developing their Expandable Variable-Autonomy Architecture (EVAA), which provides a production watchdog capability [184].

For sUAS, an RTA system currently under development for implementation is the Safe Testing of Autonomy in Complex, Interactive Environments (TACE) by the Johns Hopkins University Applied Physics Lab (JHU/APL) [145]. This system can provide a variable RTA watchdog for both onboard and offboard controls as well as interfaces for Live-Virtual-Constructive (LVC). The TACE system is currently being installed on a 412th Test Wing aircraft for initial capability demonstration and evaluation for testing of autonomy. The 412 Test Wing has recently purchased three Swift Radioplanes Lynx [186] aircraft that will be modified with a watchdog controller and an autonomy processor for demonstration of the RTA and LVC capabilities as the baseline for their Services Based Testing of Autonomy efforts.

This TACE RTA approach will provide the ability to manage, via a watchdog, the autonomous behavior of the vehicles in live flight to demonstrate the ability to mitigate violations of the watchdog. The watchdog will provide the interface between the native autopilot and the other control options (autonomy processor, automated on-board remediation, or ground control station). The native sUAS radio frequency (RF) Controller will be able to command the autopilot directly, overriding the watchdog inputs, as a redundant backup safety control. Figure 5.2 shows an example implementation of this approach. The TACE implementation will provide RTA watchdog coverage to keep the vehicles in a safe configuration. Initial capabilities for the TACE RTA implementation include protections and mitigations for airspace boundary violation, aircraft limit violation, aircraft collision avoidance, and lost link. The design of the TACE RTA allows for the addition or modification of protections and mitigations as system requirements change.

5.6 TEST APPROACH

Testing of services in a block style approach from initial algorithms to full-up services on the platforms of varying capabilities is needed. For example, initial testing of a tracking service

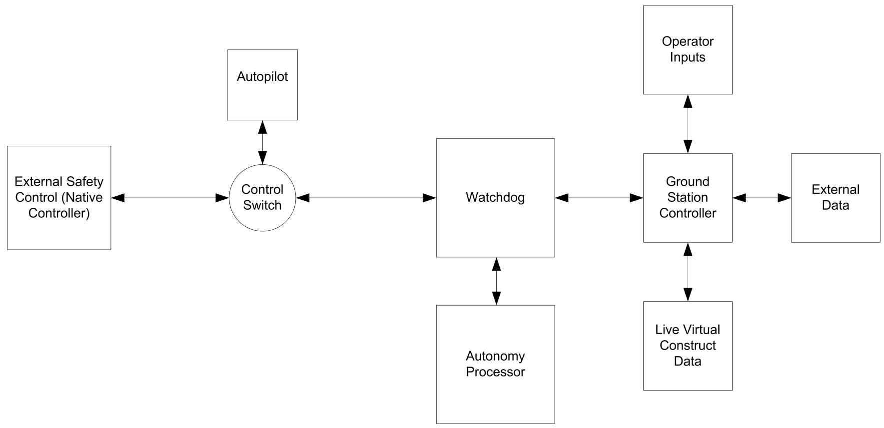

Figure 5.2: Example RTA Watchdog Implementation

without a real sensor may be performed on a Group 1 sUAS using LVC to exercise the algorithm, while sensor capabilities are being developed for a larger vehicle implementation. The new sensor would then be integrated into a larger, more capable vehicle, if required to evaluate the performance of that sensor service (i.e., tracking and target recognition capabilities) along with integration with the associated autonomy algorithm. Finally, full integration of the capability as one unique service onto another system to verify integration across platforms. These tests would show the capability of the sensor, algorithm, and integrated service on much less expensive platforms than the likely final product. As a result, when finally integrating on the final platform, the concerns become solely platform unique implementation differences and not core service functionality.

There are several key capabilities, in addition to the core SBTA architecture, required to provide a robust SBTA capability. Installation of key instrumentation parameters and recording capabilities will be critical with unique challenges, depending on the size of the vehicle. In many cases, recording capabilities will be performed within the installed processors and autopilots without additional instrumentation capability. Independent Time-Space-Position Information (TSPI) instrumentation is likely the most critical independent data requirement that will be required, depending on the system under test. Additionally, telemetry data for standard vehicle performance will be required to ensure degraded system performance is not causing poor autonomy performance.

Critical to testing autonomy, especially as the requirements for multiple vehicles and complex mission requirements increase, will be the implementation of a LVC capability. The USAF continues to consider LVC as a critical capability for both training and testing [187]. In the case of SBTA, LVC will be used to provide synthetic forces, whether air, sea, or surface and friendly (blue) or adversarial (red) forces. Additionally, the LVC capability will be able to provide synthetic threats, topography, simulated loss of communications, and simulated GPS jamming/drifting. The LVC will allow a tester to design scenarios across an increased spectrum of the state space without requiring complex and expensive range capabilities or a significant number of UASs. This approach will improve flight test safety by requiring the live flight of fewer vehicles in a clean environment while providing a complex decision space for the autonomy. Virtual entities can be simulated as part of the flight test environment and sent to the live test aircraft at real-time speeds, reducing complexity and safety in open air. Several capabilities are in different stages of development for this type of testing.

The Test and Training Enabling Architecture (TENA) provides an interface for test control that will enable the testers to use existing simulation capabilities across the test and training environment (TRMC n.d.). These TENA environment models could be made available for the LVC environment. The Navy has developed their Joint Integrated Mission Model (JIMM) that has been used on other TACE efforts for integration of synthetic forces into system demonstrations (NAWC-AD n.d.). The Boeing Company and USAF have developed the Advanced Framework for Simulation, Integration, and Modeling (AFSIM), which has been used in numerous forms and can provide synthetic force generation capability for SBTA integration [190].

Control of the autonomous systems will be performed initially using the JHU/APL developed TACE controller on a Linux laptop. However, over time, TACE could be integrated with other control capabilities with some additional development. Currently, AFRL is using the Vigilant Spirit Control System (VSCS), which has been released to numerous commercial companies [191]. The VSCS is also being interfaced with the UxAS development by AFRL, and it provides many unique capabilities for future SBTA integration.

5.7 V&V APPROACH

The key to SBTA providing a rapid testing focused on the autonomy services is well-defined hardware and software interfaces. The UxAS architecture for hardware and software integration will provide a known interface for any user to develop to, and test against, in a lab environment. Integration of the UxAS architecture and software backbone onto a heterogeneous fleet of aircraft will provide a baseline set of aircraft for testing. A series of initial testing and system verifications will be required prior to any vehicle being ready to be used as an autonomy test bed.

First, the basic aircraft will need to be verified and validated against baseline expectations. Depending on the complexity of known characteristics of a given vehicle the scope of baseline testing could range from minimal to complex. Once this initial characterization is completed, this known aircraft baseline will exist for comparison against new capabilities. These data are also critical for the implementation of autonomy services since many algorithms require knowledge of current vehicle capabilities.

Once the base aircraft has been verified, the UxAS or other system architecture, whether software or hardware, can be installed. The integrated hardware and software will then be tested to verify basic integration functionality to include basic aircraft control and communications. At the completion of this stage of testing, the system will provide an UxAS compliant test platform for integration of new autonomy services. The vehicle will then have baseline integrated performance information and will be considered an SBTA compliant aircraft.

A new autonomy service, whether software, hardware, or a combination, can then be integrated into the UxAS SBTA aircraft for service testing. This services testing will have two test phases. The first phase will be a verification of proper UxAS interface with the SBTA aircraft. This testing will verify that the communications paths between the newly installed services and the SBTA aircraft are functioning as demonstrated in the lab environment. Once the UxAS interface testing is completed, autonomy performance testing can be completed. Figure 5.3 shows the SBTA Autonomy V&V process from basic aircraft verification to autonomy performance testing.

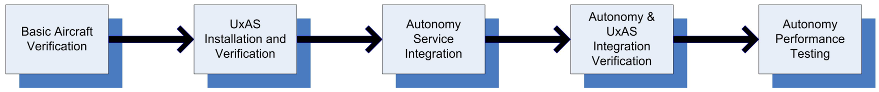

Figure 5.3: SBTA Autonomy V&V Process

5.8 IMPLEMENTATION

The Emerging Technologies Combined Test Force (ET-CTF) at the 412th Test Wing, Edwards AFB, CA in conjunction with the Air Force Research Lab Autonomy Division (AFRL/RQQA), Wright-Patterson AFB, OH is beginning to develop and acquire systems and capabilities necessary to implement SBTA.

The ET-CTF has provided three Swift Radioplanes Lynx aircraft to JHU/APL for modification and initial testing of their TACE system. The initial implementation will include use of existing JHU/APL autonomy capabilities installed on a Raspberry Pi processor, and the TACE watchdog and LVC on a second Raspberry Pi. This effort, funded by Test Resource Management Center (TRMC), will provide an integrated autonomy and watchdog capability on the Lynx vehicles to enable initial TACE LVC and watchdog functionality verification. The program will culminate in the Summer of 2018 with a multiple-vehicle single-operator demonstration of the LVC and watchdog implementation. The initial development and integration will enable improvements and upgrades of TACE watchdog (RTA) and TACE LVC into services that are an integral part of the UxAS architecture. The capability will also provide baseline development requirements for integration on other sUAS.

Figure 5.4 shows the vehicles currently planned or under consideration for SBTA integration, from left to right: Swift Radioplanes Lynx [186], UAV Factory Penguin [192], and Arcturus T-20 [193]. Table 5.2 provides key characteristics of each of the vehicles. The Swift Radioplanes Lynx Aircraft provide a low-cost, simple vehicle that is being used for initial TACE integration and evaluation. The Lynx offers the ability to do quick initial checkouts of autonomy algorithms without complex vehicle integration. The Lynx has limited sensor capability so the majority of testing will rely on LVC simulation. The UAV Factory Penguin provides a more robust lightweight test vehicle capability with larger and more robust sensor capability than the Lynx. The Penguin

Figure 5.4: Potential SBTA Test Articles

Table 5.2: Key Characteristics of Potential SBTA Test Vehicles

	Lynx	Penguin-B	Penguin-BE	Arcturus T-20
Length	N/A[1]	7.45 ft	7.45 ft	9.5 ft
Wingspan	7.5 ft	10.82 ft	10.82 ft	17.5 ft
Max Airspeed	N/A	70 Kts	70 Kts	75 Kts
Cruise Speed	31.28 Kts	42 Kts	42 Kts	N/A
Max Altitude	N/A	N/A	N/A	15,000 ft
Max Endurance	3 hrs	20+ hrs	1.8 hrs	10-20 hrs
Empty Weight	6.6 lbs	22 lbs	32.8 lbs	105 lbs
Payload	1.6 lbs	22 lbs	14.5 lb	75 lbs
Fuel	Battery	Auto Fuel	Battery	Auto Fuel
Launch	Hand	Catapult, Runway	Catapult, Runway	Catapult
Recovery	Deep Stall	Runway	Runway	Belly

[2]N/A = Not Available

is offered in both electric and fuel configurations. The electric configuration minimizes the need to deal with fuel transportation and storage but has limited endurance. Both electric and fuel powered Penguins will be used. The Arcturus T-20 provides a larger vehicle payload, endurance, and altitude capability than the Lynx or Penguin. The Arcturus has a significantly larger payload capability and power capability to enable a larger range of autonomy payload service testing. The development of the capabilities on these sUAS will enable integration onto future larger, faster, and more capable unmanned systems beyond sUAS.

The initial test capability with the Lynx vehicles is planned for testing Summer of 2018. Follow-on integration of TACE and other services with UxAS and onto other platforms is planned for the $2018 - 2020$ timeframe, with planned integration onto the Penguin and Arcturus, or similar, platforms. Future integration onto larger and faster vehicles are still under discussion and consideration with potential customers.

Using sUAS provides a significant cost advantage over the existing military aircraft fleet. First, the cost of sUAS operations is significantly lower than the thousands-of-dollars-per-hour of existing military aircraft [177]. Since most sUAS can provide similar flight duration on either battery power (minimal electrical charge costs) or a few pounds of fuel (tens-of-dollars-per-hour) the cost advantage is significant. Second, sUAS provide a simple and modifiable system design that doesn't require the same significant software and hardware integration efforts necessary to support airworthiness requirements of existing USAF manned and unmanned aircraft. Because the sUAS are low cost vehicles that are not being integrated into the USAF fleet, the level of airworthiness requirements and data to support airworthiness are significantly less. Additionally, the implementation of the RTA watchdog allows for safe reversion to a baseline aircraft in the event of poor autonomy integration. The flexibility from sUAS implementation of SBTA is expected to provide implementation, cost, and schedule advantages over existing platforms.

5.9 CONCLUSIONS

Services-Based Testing of Autonomy, upon successful implementation, is expected to provide a cost-effective, scalable, and efficient means to test the growing needs for autonomy services that are being developed across the industry. The use of a common open architecture should allow for integration into diverse platforms rapidly for quick assessment of autonomy services. The cost to perform testing on sUAS is anticipated to be significantly less than on larger vehicles, and allows for rapid changes to hardware and software in a more flexible manner than traditional test vehicles. Implementation and future studies utilizing SBTA on sUAS will provide the data necessary to enable verification of the viability of the SBTA approach for testing of autonomy. Additionally, while this approach is primarily focused on sUAS implementation for cost-effective testing of vehicle operations, the same approach can be implemented directly on larger UASs or even manned aircraft with a subset of capability implementation via autonomy.

CHAPTER 6

CONCLUSIONS AND REMARKS

This dissertation provides an approach for autonomous UAV control and testing with details on an initial implementation for Services Based Testing of Autonomy and an expanded POMDP control schema. We have provided a high level architecture approach for autonomy implementation, two upgrades to our POMDP algorithm, and an approach for testing autonomy using sUAS.

In Chapter 2, we presented a high level architecture for control of UAS for autonomous mission. The design approach provided a modular architecture that enables safe and effective control. We also provided a review of the current state of path planning and safety controls that could be implemented in such an architecture and identified areas that needed improvement. This architecture and review provided the baseline for future efforts in path planning and system design for implementation in the support of TEVV of autonomy.

In Chapter 3, we extended previous efforts in the development of POMDP path planning by implementing controls with a limited FOV sensor. The inclusion of a limited FOV sensor challenges the POMDP algorithms estimation capability compared to a system that assumed it could always see the target(s). The results of this implementation showed the robust capability of the POMDP to be able to maintain tracking of the moving target, even during extended periods of non-observation periods. Lower altitudes showed limited effectiveness, but larger altitudes provided expected tracking results.

In Chapter 4, we implemented a fuel efficient cost function to our existing POMDP algorithm. The baseline cost function focuses on minimizing tracking errors. The addition of a fuel cost into the cost function provides the ability to improve endurance without significantly increasing tracking errors, for most altitudes. The addition of the fuel efficient cost function provided a variable impact, up to 10% increase in endurance. Lower altitude results showed a larger impact on tracking errors with increased fuel efficiency, and generally showed undesirable results at the lowest altitude. Future expansion to include a gimbaled sensor is expected to improve tracking performance, especially at low altitudes.

In Chapter 5, we provided a system architecture and approach for utilizing sUAS for TEVV of Autonomy called "Services Based Testing of Autonomy." This approach provides the capability to modularize the systems and enable a common system architecture across a heterogeneous fleet of sUAS. The low-cost and low-complexity of implementation on sUAS provides a means for testing of autonomy in a rapid, repeatable environment. The approach is currently being implemented by the Emerging Technologies Combined Test Force at Edwards AFB.

The implementation of autonomy services into sUAS will provide a capability to integrate and test a diverse set of autonomy algorithms and sensors, including the POMDP algorithms presented in Chapters 3 and 4. The future viability of the SBTA approach will be dependent upon the UxAS system architecture along with the integration of TACE watchdog and LVC capabilities. Implementation and testing of autonomy is a rapidly growing area of concern. The research presented within this dissertation provides improvements to an existing autonomy algorithm and also provides an implementation approach to enable repeatable and dependable testing of autonomy capabilities.

BIBLIOGRAPHY

[1] Teal Group Predicts Worldwide UAV Market Will Total $91 Billion in Its 2014 UAV Market Profile and Forecast. Available online: http://www.tealgroup.com/index.php/about-teal-group-corporation/press-releases/118-2014-uav-press-release (accessed on 19 August 2015).

[2] Valvanis, K.; Vachtsevanos, G. Future of Unmanned Aviation. In *Handbook of Unmanned Aerial Vehicles*; Springer: Dordrecht, The Netherlands, 2015; pp. 2993–3009.

[3] DARPA Collaborative Operations in a Denied Environment (CODE). DARPA-BAA-14-33. Available online: https://www.fbo.gov/spg/ODA/DARPA/CMO/DARPA-SN-14-28/listing.html (accessed on 30 November 2015).

[4] DARPA Distributed Battle Management Program. DARPA-BAA-14-17. Available online: https://www.fbo.gov/spg/ODA/DARPA/CMO/DARPA-BAA-14-17/listing.html (accessed on 30 November 2015).

[5] Otto, R.P. *Air Force ISR 2023: Delivering Decision Advantage*; Headquarters United States Air Force: Washington, DC, USA, 2013.

[6] Dempsey, M.E. *Intelligence, Surveillance, and Reconnaissance Joint Force 2020 White Paper*; U.S. Army: Washington, DC, USA, 2014.

[7] Seffers, G. Joint Aerial Layer Network Vision Moves Toward Reality. Available online: http://www.afcea.org/content/?q=node/11123 (accessed on 19 August 2015).

[8] Schechter, E. UAVs Could be Next Step for Electronic Warfare. Available online: http://archive.c4isrnet.com/article/20140507/C4ISRNET08/305070006/UAVs-could-next-step- electronic-warfare (accessed on 19 August 2015).

[9] Cevik, P.; Kocaman, I.; Akgul, A.; Akca, B. The Small and Silent Force Multiplier: A Swarm UAV—Electronic Attack. *J. Intell. Robot. Syst.* **2013**, *70*, 595–608.

[10] Callam, A. Drone Wars: Armed Unmanned Aerial Vehicles. *Int. Aff. Rev.* **2010**, *18*, 3.

[11] Lockheed/Piasecki Team Tackles Cargo UAV. 2014. Available online: http://aviationweek.com/awin/lockheedpiasecki-team-tackles-cargo-uav (20 August 2015).

[12] Myers, M. New funds to aid coast guard in adopting a UAV. *Navy Times*, 25 April 2015.

[13] Reyes, H.; Gellerman, N.; Kaabouch, N. A Cognitive Radio System for Improving the Reliability and Security of UAS/UAV Networks. In Proceedings of the 2015 IEEE Aerospace Conference, Big Sky, MT, USA, 7–14 March 2015; pp. 1–9.

[14] Ogan, R. Educating the Next Generation Engineers for Unmanned Aircraft Systems Applications and Challenges. In Proceedings of the SoutheastCon 2015, Ft. Lauderdale, FL, USA, 9–12 April 2015.

[15] Boskovic, J.D.; Prasanth, R.; Mehra, R.K. A Multilayer Control Architecture for Unmanned Aerial Vehicles. In Proceedings of the American Control Conference, Anchorage, AK, USA, 8–10 May 2002; pp. 1825–1830.

[16] Dantzig, G.; Fulkerson, R.; Johnson, S. Solution of a Large-Scale Traveling-Salesman Problem. *J. Oper. Res. Soc. Am.* **1954**, *2*, 393–410.

[17] Garey, M.R.; Johnson, D.S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*; Freeman: New York, NY, USA, 1985.

[18] Glover, F. Tabu Search: A Tutorial. *Interfaces* **1990**, *20*, 74–94.

[19] Glover, F.; Laguna, M. *Tabu Search*; Kluwer Academic Publishers: Boston, MA, USA, 1997.

[20] Ryan, J.; Bailey, T.; Moore, J. Reactive Tabu Search in Unmanned Aerial Reconnaissance Simulations. In Proceedings of the IEEE Winter Simulation Conference, Washington, DC, USA, 13–16 December 1998.

[21] Wang, Z.; Liu, Q.; Tao, H.; Li, J. Multiple Task Planning Based on TS Algorithm for Multiple Heterogenous Unmanned Aerial Vehicles. In Proceedings of the IEEE Chinese Guidance, Navigation and Control Conference, Yantai, China, 8–10 August 2014.

[22] Zhao, J.; Zhao, J. Study on Multi-UAV Task clustering and Task Planning in Cooperative Reconnaissance. In Proceedings of the International Conference on Intelligent Human-Machine Systems and Cybernetics, Hangzhou, China, 26–27 August 2014.

[23] Dijkstra, E. A Note on Two Problems in Connexion with Graphs. *Numer. Math.* **1959**, *1*, 269–271.

[24] Meng, B.; Gao, X.; Wang, Y. Multi-mission Path Re-planning for Multiple Unmanned Aerial Vehicles Based on Unexpected Events. In Proceedings of the International Conference on Intelligent Human-Machine Systems and Cybernetics, Hangzhou, China, 26–27 August 2009; pp. 423–426.

[25] Bertuccelli, L.; Choi, H.; Cho, P.; How, J. Real-time Multi-UAV Task Assignment in Dynamic and Uncertain Environments. In Proceedings of the AIAA Guidance, Navigation and Control Conference, Chicago, IL, USA, 10–13 August 2009.

[26] Leary, S.; Deittert, M.; Bookless, J. Constrained UAV Mission Planning: A Comparison of Approaches. In Proceedings of the IEEE International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2002–2009.

[27] Ragi, S.; Chong, E.K.P. Decentralized Guidance Control of UAVs with Explicit Optimization of Communications. *J. Intell. Robot. Syst.* **2014**, *73*, 811–822.

[28] Cummings, M.; How, J.; Whitten, A.; Toupet, O. The Impact of Human-Automation Collaboration in Decentralized Multiple Unmanned Vehicle Control. *Proc. IEEE* **2012**, *100*, 660–671.

[29] Kopeikin, A.; Ponda, S.; Johnson, L.; How, J. Multi-UAV Network Control Through Dynamic Task Allocation: Ensuring Data-Rate and Bit-Error-Rate Support. In Proceedings of the IEEE Globecom Workshop, Anaheim, CA, USA, 3–7 December 2012; pp. 1579–1584.

[30] Tong, H.; Wen, W.; Chang, H.; Yong, X. Path Planning of UAV Based on Voronoi Diagram and DPSO. In Proceedings of the International Workshop on Information and Electronics Engineering (IWIEE), Harbin, China, 10–11 March 2012; pp. 4198–4203.

[31] Mattingley, J.; Wang, Y.; Boyd, S. Receding Horizon Control. *Control Syst. IEEE* **2012**, *31*, 52–65.

[32] Kuwata, Y.; Richards, A.; Schouwenaars, T.; How, J. Decentralized Robust Receding Horizon Control for Multi-vehicle Guidance. In Proceedings of the American Control Conference, Minneapolis, MN, USA, 14–16 June 2006.

[33] Kuwata, Y.; How, J. Robust Cooperative Decentralized Trajectory Optimization using Receding Horizon MILP. In Proceedings of the American Control Conference, New York, NY, USA, 11–13 July 2007.

[34] Xiao, X.; Dong, Z.; Wu, J.; Duan, H. A Cooperative Approach to Multiple UAVs Searching for Moving Targets Based on a Hybrid of Virtual Force and Receding Horizon. In Proceedings of the IEEE 10th International Conference on Industrial Informatics, Beijing, China, 25–27 July 2012; pp. 1228–1233.

[35] Peng, H.; Su, F.; Bu, Y.; Zhang, G.; Shen, L. Cooperative Area Search for Multiple UAVs based on RRT and Decentralized Receding Horizon Optimization. In Proceedings of the 7th Asian Control Conference, Hong Kong, China, 27–29 August 2009.

[36] Schouwenaars, T.; How, J.; Feron, E. Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees. In Proceedings of the AIAA Guidance, Navigation and Control Conference, Providence, RI, USA, 16–19 August 2004.

[37] Miller, S.; Harris, Z.; Chong, E.K.P. A POMDP Framework for Coordinated Guidance of Autonomous UAVs for Multitarget Tracking. *EURASIP J. Adv. Signal Process.* **2009**, *2009*, 1–17.

[38] Ragi, S.; Tan, C.; Chong, E.K.P. Feasibility Study of POMDP in Autonomous Amphibious Vehicle Guidance. In Proceedings of the IFAC Intelligent Autonomous Vehicles Symposium, Gold Coast, Australia, 26–28 June 2013.

[39] Ragi, S.; Chong, E.K.P. UAV Path Planning in a Dynamic Environment via Partially Observable Markov Decision Process. *IEEE Trans. Aerosp. Electron. Syst.* **2013**, *49*, 2397–2412.

[40] Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the IEEE International Conference on Neural Networks, Piscataway, NJ, USA, 27 November–1 December 1995; pp. 1942–1948.

[41] Wang, G.; Li, Q.; Guo, L. Multiple UAVs Routes Planning Based on Particle Swarm Optimization Algorithm. In Proceedings of the 2nd International Symposium on Information Engineering and Electronic Commerce (IEEC), Ternopil, Ukraine, 23–25 July 2010.

[42] Kenefic, R. Finding Good Dubins Tours for UAVs Using Particle Swarm Optimization. *J. Aerosp. Comput. Inf. Commun.* **2008**, *5*, 47–56.

[43] Zhao, J.; Zhao, J. Target Distributing of Multi-UAVs Cooperative Attack and Defend Based on DPSO Algorithm. In Proceedings of the Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics, Hangzhou, China, 26–27 August 2014; pp. 396–400.

[44] Alejo, D.; Cobano, A.; Heredia, G.; Ollero, A. Collision-free Trajectory Planning Based on Maneuver Selection-Particle Swarm Optimization. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015.

[45] Roberge, V.; Tarbouchi, M.; Labonte G. Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV Path Planning. *IEEE Trans. Ind. Inform.* **2013**, *9*, 132–141.

[46] Sahingoz, O.K. Flyable Path Planning for a Multi-UAV System with Genetic Algorithms and Bezier Curves. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 28–31 May 2013; pp. 41–48.

[47] Sahingoz, O.K. Generation of Bezier Curve-Based Flyable Trajectories for Multi-UAV Systems with Parallel Genetic Algorithm. *J. Intell. Robot. Syst. Theory Appl.* **2014**, *74*, 499–511.

[48] Cekmez, U.; Ozsiginan, M.; Sahingoz, O.K. Adapting the GA Approach to Solve Traveling Salesman Problems on CUDA Architecture. In Proceedings of the IEEE International Symposium on Computational Intelligence and Informatics (CINTI 2013), Budapest, Hungary, 19–21 November 2013; pp. 423–428.

[49] Cekmez, U.; Ozsiginan, M.; Sahingoz, O.K. A UAV Path Planning with Parallel ACO Algorithm on CUDA Platform. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 27–30 May 2014; pp. 347–354.

[50] Cheng, Z.; Sun, Y.; Liu, Y. Path Planning Based on Immune Genetic Algorithm for UAV. In Proceedings of the International Conference on Electric Information and Control Engineering (ICEICE), Wuhan, China, 15–17 April 2011; pp. 590–593.

[51] Price, I.; Lamont, G. GA Directed Self-Organized Search and Attack UAV Swarms. In Proceedings of the Winter Simulation Conference, Monterey, CA, USA, 3–6 December 2006; pp. 1307–1315.

[52] Pehlivanoglu, Y. A New Vibrational Genetic Algorithm Enhanced with a Voronoi Diagram for Path Planning of Autonomous UAV. *Aerosp. Sci. Technol.* **2012**, *16*, 47–55.

[53] Sonmez, A.; Kocyigit, E.; Kugu, E. Optimal Path Planning for UAVs Using Genetic Algorithm. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015.

[54] Geng, L.; Zhang, Y.F.; Wang, J.J.; Fuh, J.Y.H.; Teo, S.H. Cooperative Task Planning for Multiple Autonomous UAVs with Graph Representation and Genetic Algorithm. In Proceedings of the 10th IEEE International Conference on Control and Automation (ICCA), Hangzhou, China, 12–14 June 2013.

[55] Kirkpatrick, S.; Gelett, C.D.; Vecchi, M.P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 621–630.

[56] Cerny, V. A Thermodynamic Approach to the Traveling Salesman Problem: An Efficient Simulation. *J. Optim. Theory Appl.* **1985**, *45*, 41–51.

[57] Turker, T.; Sahingoz, O.K.; Yilmaz, G. 2D Path Planning for UAVs in Radar Threatening Environment using Simulated Annealing Algorithm. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015.

[58] Doringo, M. Optimization, Learning and Natural Algorithms. Ph.D. Thesis, Politecnico di Milano, Milan, Italy, 1992.

[59] Fallahi, K.; Leung, H.; Chandana, S. An Integrted ACO-AHP Approach for Resource Management Optimization. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, San Antonio, TX, USA, 11–14 October 2009.

[60] Duan, H.; Zhang, X.; Wu, J.; Ma, G. Max-Min Adaptive Ant Colony Optimization Approach to Multi-UAVs Coordinated Trajectory Replanning in Dynamic and Uncertain Envirionments. *J. Bionic Eng.* **2009**, *6*, 161–173.

[61] Qu, Y.; Zhang, Y.; Zhang, Y. A UAV Solution of Regional Surveillance Based on Pheromones and Artificial Potential Field Theory. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015.

[62] Zhang, Y.; Xiao, Y. A Patrolling Scheme in Wireless Sensor and Robot Networks. In Proceedings of the IEEE Conference on Computer Communications Workshops, Shanghai, China, 10–15 April 2011; pp. 513–518.

[63] Zhang, Y.; Xiao, Y. Digital Pheromone Based Patrolling Algorithm in Wireless Sensor and Actuator Networks. In Proceedings of the IEEE Consumer Communications and Networking Conference, Las Vegas, NV, USA, 11–14 January 2013; pp. 496–501.

[64] Shang, K.; Karungaru, S.; Feng, Z.; Ke, L.; Terada, K. A GA-ACO Hybrid Algorithm for the Multi-UAV Mission Planning Problem. In Proceedings of the Technologies (ISCI-International Symposium on Communications and Information T), Incheon, Korea, 24–26 September 2014; pp. 243–248.

[65] Krishnamoorthy, K.; Casbeer, D.; Pachter, M. Minimum Time UAV Pursuit of a Moving Ground Target using Partial Information. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015.

[66] Krishnamoorthy, K.; Casbeer, D.; Chandler, P.; Pachter, M.; Darbha, S. UAV Search & Capture of a Moving Ground Target under Delayed Information. In Proceedings of the IEEE Conference on Decision and Control, Mauai, Hawaii, USA, 10–13 December 2012.

[67] Rasmussen, S.; Kingston, D. Development and Flight Test of an Area Monitoring System Using Unmanned Aerial Vehicles and Unattended Ground Sensors. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015.

[68] Moon, S.; Yang, K.; Gan, S.; Shim, D. Decentralized Information-theoretic Task Assignment for Searching and Tracking of Moving Targets. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015.

[69] Sun, A.; Liu, H. Cooperative UAV Search for Moving Targets Using a Modified Diffusion Uncertainty Model. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Chicago, IL, USA, 10–13 August 2009.

[70] Frew, E.; Lawrence, D.; Morris, S. Coordinated Standoff Tracking of Moving Targets Using Lyapunov Guidance Vector Fields. *J. Guid. Control Dyn.* **2008**, *31*, 290–306.

[71] Summers, T.; Akella, M. Coordinated Standoff Tracking of Moving Targets: Control Laws and Information Architectures. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Honolulu, HI, USA, 18–21 August 2008.

[72] Summers, T.; Akella, M.; Mears, M. Coordinated Standoff Tracking of Moving Targets: Control Lawas and Information Architectures. *J. Guid. Control Dyn.* **2009**, *32*, 56–69.

[73] Geyer, C. Active Target Search from UAVs in Urban Environments. In Proceedings of the IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008.

[74] Bertuccelli, L.; How, J. UAV Search for Dynamic Targets with Uncertain Motion Models. In Proceedings of the IEEE Conference on Decision & Control, San Diego, CA, USA, 13–15 December 2006.

[75] Ragi, S.; Chong, E.K.P. UAV Guidance Algorithms via Partially Observable Markov Decision Processes. In *Handbook of Unmanned Aerial Vehicles*; Valvanis, K., Vachtsevanos, G., Eds.; Springer: Dordrecht, The Netherlands, 2015; pp. 1775–1810.

[76] Bertuccelli, L.F.; How, J.P. Robust UAV Search for Environments with Imprecise Probability Maps. In Proceedings of the IEEE Conference on Decision and Control, and the European Control Conference, Seville, Spain, 12–15 December 2005.

[77] Song, S.; Rodriguez, A.; Teodorescu, M. Trajectory Planning for Autonomous Nonholonomic Vehicles for Optimal Monitoring of Spatial Phenomena. In Proceedings of the In-

ternational Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015.

[78] Lee, S.; Morrison, J. Decision Support Scheduling for Maritime Search and Rescue Planning with a system of UAVs and Fuel Service Stations. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015.

[79] Zhang, C.; Pei, H. Oil Spills Boundary Tracking Using Universal Kriging and Model Predictive Control by UAV. In Proceedings of the 11th World Congress on Intelligent Control and Automation, Shenyang, China, 29 June–4 July 2014.

[80] Hu, J.; Xie, L.; Lum, K.; Xu, J. Multiagent Information Fusion and Cooperative Control in Target Search. *IEEE Trans. Control Syst. Technol.* **2013**, *21*, 1223–1235.

[81] Hirsch, M.; Schroeder, D. On the Decentralized Cooperative Control of Multiple Autonomous Vehicles. In *Handbook of Unmanned Aerial Vehicles*; Valvanis, K., Vachtsevanos, G., Eds.; Springer: Dordrecht, The Netherlands, 2015; pp. 1775–1810.

[82] Hirsch, M.; Schroeder, D. Dynamic Decentralized Cooperative Control of Multiple Autonomous Vehicles with Multiple Tasks for Urban Operations. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Minneapolis, MN, USA, 13–16 August 2012.

[83] Wu, P.; Campbell, D.; Merz, T. On-Board Multi-Objective Mission Planning for Unmanned Aerial Vehicles. In Proceedings of the IEEE Aerospace Conference, Piscataway, NJ, USA, 7–14 March 2009; pp. 1–10.

[84] Ilaya, O. Multi Objective Decentralized Model Predictive Control for Cooperative Multi-UAV Systems. In Proceedings of the AIAA Guidance, Navigation and Control Conference, Hilton Head, SC, USA, 20–23 August 2007.

[85] Ilaya, O.; Bil, C.; Evans, M. Distributed and Cooperative Decision Making for Multi-UAV Systems with Applicaiton to Collaborative Electronic Warfare. In Proceedings of the AIAA

Aviation Technology, Integration and Operations Conference, Belfast, Northern Ireland, 18–20 September 2007.

[86] Peng, X.; Xu, D.; Yan, W. Intelligent Flight for UAV via Integration of Dynamic MOEA, Bayesian Network and Fuzzy Logic. In Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference, Orlando, Fl, USA, 12–15 December 2011.

[87] Chapman, A.; Mesbahi, M. UAV Swarms: Models and Effective Interfaces. In *Handbook of Unmanned Aerial Vehicles*; Valvanis, K., Vachtsevanos, G., Eds.; Springer: Dordrecht, The Netherlands, 2015; pp. 1775–1810.

[88] Barca, J.; Sekercioglu, Y. Swarm robotics reviewed. *Robotica* **2013**, *31*, 345–359.

[89] Alighanbari, M.; How, J. An Unbiased Kalman Consensus Algorithm. In Proceedings of the American Control Conference, Minneapolis, MN, USA, 14–16 June 2006.

[90] Brunet, L.; Choi, H.; How, J. Consensus-Based Auction Approaches for Decentralized Task Assignment. In Proceedings of the AIAA Guidance Navigation and Control Conference, Honolulu, HI, USA, 18–21 August 2008.

[91] Choi, H.; Brunet, L.; How, J. Consensus-Based Decentralized Auctions for Robust Task Allocation. *IEEE Trans. Robot.* **2009**, *25*, 912–926.

[92] Choi, H.; Whitten, A.; How, J. Decentralized Task Allocation for Heterogeneous Teams with Cooperation Constraints. In Proceedings of the 2010 American Control Conference, Baltimore, MD, USA, 30 June–2 July 2010.

[93] Ren, W.; Beard, R.; Atkins, E. A Survey of Consensus Problems in Multi-agent Coordination. In Proceedings of the 2005 American Control Conference, Portland, OR, USA, 8–10 June 2005.

[94] Ren, W.; Beard, R. *Distributed Consensus in Multi-vehicle Cooperative Control*; Springer-Verlag: London, UK, 2008.

[95] Ren, W.; Beard, R. Consensus Seeking in Multiagent Systems under Dynamically Changing Interaction Topologies. *IEEE Trans. Autom. Control* **2005**, *50*, 655–661.

[96] Sujit, P.; Kingston, D.; Beard, R. Cooperative Forest Fire Monitoring Using Multiple UAVs. In Proceedings of the 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, 12–14 December 2007.

[97] Kingston, D.; Beard, R. Decentralized Perimeter Surveillance Using a Team of UAVs. *IEEE Trans. Robot.* **2008**, *24*, 1394–1404.

[98] Ou, W.; Zou, F.; Xu, X.; Gao, Z. Targets Assignment for Cooperative Multi-UAVs Based on Chaos Optimization Algorithm. In Proceedings of the 9th International Conference for Young Computer Scientists, Hunan, China, 18–21 November 2008.

[99] Zhang, B.; Mao, Z.; Liu, W.; Liu, J.; Zheng, Z. Cooperative and Geometric Learning for Path Planning of UAVs. In Proceedings of the 2013 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 18–21 May 2013.

[100] Clark, M.; Koutsoukos, X.; Kumar, R.; Lee, I.; Pappas, G.; Pike, L.; Porter, J.; Sokolsky, O. A *Study on Run Time Assurance for Complex Cyber Physical Systems*; Air Force Research Lab: Dayton, OH, USA, 2013.

[101] Zhang, X.; Clark, M.; Rattan, K.; Muse, J. Controller Verification in Adaptive Learning Systems Towards Trusted Autonomy. In Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems, Seattle, WA, USA, 14–16 April 2015.

[102] Schierman, J.; DeVore, M.; Cooper, J.; Richards, N.; Gandhi, N.; Horneman, K.; Smolka, S.; Stoller, S.; Clark, M. Run Time Assurance for Complex Autonomy. In Proceedings of the Safe and Secure Systems and Software Symposium, Dayton, OH, USA, 9–11 June 2015.

[103] Skoog, M. Expandable Variable-Autonomy Architecture. NASA Fact Sheet, 2015. Available online: http://www.nasa.gov/centers/armstrong/news/FactSheets/index.html (accessed on 19 August 2015).

[104] Ho, N.; Koltai, K.; Masequesmay, G.; Cals, S.; Sadler, G.; Lyons, J.; Cancanindin, A.; Johnson, W.; Skoog, M. An Ethnographic-Based Model for Trust Development in Auto-GCAS. In Proceedings of the American Psychological Associate Annual Convention, Toronto, ON, Canada, 6–9 August 2015.

[105] Niedober, D.; Ho, N.; Koltai, K.; Masequesmay, G.; Skoog, M.; Cacanindin, A.; Johnson, W.; Lyons, J. Influence of Cultural, Organizational, and Automation Capability Factors on Human-Automation Trust: A Case Study of Auto-GCAS Engineers. In Proceedings of the International Conference on Human Computer Interaction, Crete, Greece, 22–27 June 2014.

[106] Koltai, K.; Ho, N.; Masequesmay, G.; Niedober, D.; Skoog, M.; Cacanindin, A.; Johnson, W.; Lyons, J. Influence of Cultural, Organizational, and Automation Capability on Human Automation Trust: A Case Study of Auto-GCAS Experimental Test Pilots. In Proceedings of the International Conference on Human Computer Interaction in Aerospace, Santa Clara, CA, USA, 30 July–1 August 2014.

[107] Koltai, K.; Ho, N.; Masequesmay, G.; Niedober, D.; Skoog, M.; Cacanindin, A.; Johnson, W.; Lyons, J. An Extended Case Study Methodology for Investigating Influence of Cultural, Organizational, and Automation Factors on Human-Automation Trust. In Proceedings of the ACM-CHI 2014 Conference, Toronto, ON, Canada, 26 April–1 May 2014.

[108] Bice, G.; Skoog, M.; Howard, J. Aircraft Ground Collision Avoidance and Autorecovery System Device. U.S. Patent 4,924,401, 8 May 1990.

[109] Skoog, M.; Prosser, K. *Advanced Fighter Technology Integration/F-16 Automatic Ground Collision Avoidance System Evaluation*; AFFTC-TR-99-28; Wright-Patterson Air Force Base: Dayton, OH, USA, 2000.

[110] Moore, D.; Schlappi, K. *USAF F-16 Block 40/50 Test and Evaluation for the Automatic Ground Collision Avoidance System (Auto GCAS) and Pilot Activated Recovery System (PARS)*; 412TW-TR-13-04; Wright-Patterson Air Force Base: Dayton, OH, USA, 2013.

[111] Sorokowski, P. *Automatic Ground Collision Avoidance System Fighter Risk Reduction Project*; AFFTC-TIM-10-05; Wright-Patterson Air Force Base: Dayton, OH, USA, 2010.

[112] Swihart, D.E.; Barfield, A.F.; Griffin, E.M.; Lehmann, R.C.; Whitcomb, S.C.; Flynn, B.; Skoog, M.A.; Processor, K.E. Automatic Ground Collision Avoidance System design, integration, & flight test. *Aerosp. Electron. Syst. Mag.* **2011**, *26*, 4–11.

[113] Norris, G. Ground Collision Avoidance System "Saves" First F-16 in Syria. 5 February 2015. Available online: http://aviationweek.com/defense/ground-collision-avoidance-system-saves-first-f-16-syria (accessed on 11 October 2015).

[114] Sorokowski, P.; Skoog, M.; Burrows, S.; Thomas, S. *Small UAV Automatic Ground Collision Avoidance System Design Considerations and Flight Test Results*; NASA/TM-2015-218732; NASA: Washington, DC, USA, 2015.

[115] Scherer, S.; Singh, S.; Chamberlain, L.; Elgersma, M. Flying Fast and Low Among Obstacles: Methodology and Experiments. *Int. J. Robot. Res.* **2008**, *27*, 549–574.

[116] Kobilarov, M. Cross-entropy Motion Planning. *Int. J. Robot. Res.* **2012**, *31*, 855–871.

[117] Hrabar, S. Reactive Obstacle Avoidance for Rotorcraft UAVs. In Proceedings of the Intelligent Robots and Systems Conference, San Francisco, CA, USA, 25–30 September 2011.

[118] Wood, D. Collision Avoidance System and Method Utilizing Variable Surveillance Envelope. U.S. Patent 6,804,607, 12 October 2004.

[119] Noth, K. Modeling and Simulation of a Ground Based Sense and Avoid Architecture for Unmanned Aircraft System Operations. In Proceedings of the Integrated Communications, Navigation and Surveillance Conference, Herndon, VA, USA, 11–13 May 2011.

[120] Ottman, M. Army Ground Based Sense and Avoid (GBSAA): Enables Unmanned Aircraft Flight in the National Airspace. Available online: http://www.army.mil/article/80681/Army_Ground_Based_Sense_and_Avoid__GBSAA (accessed on 11 October 2015).

[121] Campbell, P. Challenges to using Ground Based Sense and Avoid (GBSAA) for UAS Operations. In Proceedings of the Digital Avionic Systems Conference, Williamsburg, VA, USA, 14–18 October 2012.

[122] Federal Aviation Administration. Automatic Dependent Surveillance-Broadcast (ADS-B). Available online: http://www.faa.gov/nextgen/programs/adsb/ (accessed on 11 October 2015).

[123] Lin, Y.; Saripalli, S. Sense and Avoid for Unmanned Aerial Vehicles Using ADS-B. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015.

[124] Liu, W.; Hwang, I. Probabilistic Aircraft Midair Conflict Resolution Using Stochastic Optimal Control. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 37–46.

[125] Richards, A.; How, J. Aircraft Trajectory Planning With Collision Avoidance Using Mixed Integer Linear Programming. In Proceedings of the American Control Conference, Anchorage, AK, USA, 8–10 May 2002.

[126] Patel, R.; Goulart, P.; Serghides, V. Real-Time Trajectory Generation for Aircraft Avoidance Maneuvers. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Chicago, IL, USA, 10–13 August 2009.

[127] Lin, Y.; Saripalli, S. Collision Avoidance for UAVs Using Reachable Sets. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015.

[128] Lin, Y.; Saripalli, S. Path Planning Using 3D Dubins Curve for Unmanned Aerial Vehicles. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, CA, USA, 27–30 May 2014.

[129] Shim, D.; Sastry, S. An Evasive Maneuvering Algorithm for UAVs in See-and-Avoid Situations. In Proceedings of the American Control Conference, New York City, NY, USA, 11–13 July 2007.

[130] Bareiss, D.; van den Berg, J. Reciprocal Collision Avoidance for Robots with Linear Dynamics using LQR-Obstacles. In Proceedings of the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013.

[131] Wolf, T.; Kochenderfer, M. Aircraft Collision Avoidance Using Monte Carlo Real-Time Belief Space Search. *J. Intell. Robot. Syst.* **2011**, *64*, 277–298.

[132] Temizer, S.; Kochenderfer, M.; Kaelbling, L.; Lozano-Perez, T.; Kuchar, J. Collision Avoidance for Unmanned Aircraft using Markov Decsion Processes. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Toronto, ON, Canada, 2–5 August 2010.

[133] Bai, H.; Hsu, D.; Kochenderfer, M.; Lee, W. Unmanned Aircraft Collision Avoidance Using Continuous-State POMDPs. In *Robotics: Science and Systems*; MIT Press: Cambridge, MA, USA, 2012; pp. 1–8.

[134] Jackson, J.; Boskovic, J.; Diel, D. Sensor Fusion for Sense and Avoid for Small UAS without ADS-B. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015.

[135] Williamson, T.; Spencer, N. Development and Operation of the Traffic Alert and Collision Avoidance System (TCAS). *Proc. IEEE* **1989**, *77*, 1735–1744.

[136] Wilson, M. The Use of Low-Cost Mobile Radar Systems for Small UAS Sense and Avoid. In *Sense and Avoid in UAS Research and Applications*; Angelov, P., Ed.; John Wiley and Sons: West Sussex, UK, 2012; pp. 295–336.

[137] Griffith, J.D.; Kochenderfer, M.J.; Kuchar, J.K. Electro-optical system analysis for sense and avoid. In Proceedings of the AIAA Guidance, Navigation and Control Conference, Honolulu, HI, USA, 18–21 August 2008.

[138] Ramasamy, S.; Gardi, A.; Liu, J.; Sabatini, R. A Laser Obstacle Detection and Avoidance System for Manned and Unmanned Aircraft Applicatins. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015.

[139] Angelov, P. *Sense and Avoid in UAS Research and Applications*; Angelov, P., Ed.; John Wiley and Sons: West Sussex, UK, 2012; pp. 295–336.

[140] Contarino, M. All Weather Sense and Avoid System for UASs Report Task 3.1 - Review & Analysis of Available System Technology Options and Justification for System Selection. US Office of Naval Research, 2009.

[141] Stevens, M.; Coloe, B.; Atkins, E. Platform-Independent Geofencing for Low Altitude UAS Operations. In Proceedings of the 15th Aviation Technology Integration, and Operations Conference, Dallas, TX, USA, 22–26 June 2015.

[142] Arduino Autopilot Geo-Fencing. Available online: http://plane.ardupilot.com/wiki/geofencing/ (accessed on 13 September 2015).

[143] Hayhurst, K.; Maddalon, J.; Neogi, N.; Verstynen, H. A Case Study for Assured Containment. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015.

[144] Range Commanders Council. *Flight Safety System (FSS) for Unmanned Aerial Vehicle (UAV) Operation*; Special Report; U.S. Army: Washington, DC, USA, 2008.

[145] Scheidt, D.; D'Amico, W.; Lutz, R. Safe Testing of Autonomy in Complex, Interactive Environments (TACE). *ITEA J.* **2014**, *35*, 323–331.

[146] Stansbury, R.; Tanis, W.; Wilson, T. A Technology Survey of Emergency Recovery and Flight Termination Systems for UAS. In Proceedings of the AIAA Infotech Aerospace Conference, Seattle, WA, USA, 6–9 April 2009.

[147] C. M. Eaton *et al.*, “Multiple-Scenario Unmanned Aerial System Control: A Systems Engineering Approach and Review of Existing Control Methods,” *Aerospace*, vol. 3, no. 1, pp. 1–26, Jan. 2016.

[148] S. Ragi and E. K. P. Chong, “UAV Guidance Algorithms Via Partially Observable Markov Decision Processes,” in *Handbook of Unmanned Aerial Vehicles*, K.P. Valavanis, G.J. Vachtsevanos, Eds., Netherlands: Springer, 2015, pp. 1775–1810.

[149] S. Ragi and E. K. P. Chong,“UAV path planning in a dynamic environment via partially observable Markov decision process,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 4, pp. 2397–2412, Oct. 2013.

[150] S. A. Miller *et al.*, “A POMDP framework for coordinated guidance of autonomous UAVs for multitarget tracking,” *EURASIP J. Adv. Signal Process.*, vol. 2009, 2009.

[151] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwood, MA: Artech House, 1999.

[152] Y. Bar-Shalom *et al.*, *Estimation with Applications to Tracking and Navigation*. Hoboken, NJ: Wiley-Interscience, 2001.

[153] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. Waltham, MA: Academic Press, 1988.

[154] E. K. P. Chong *et al.*, “Partially observable Markov decision process approximations for adaptive sensing,” *Discrete Event Dynamic Systems*, vol. 19, 2009, pp. 377–422.

[155] R. Bellman, *Dynamic Programming*. Princeton, NJ: Princeton University Press, 1957.

[156] C. Kreucher *et al.*, “Efficient methods of non-myopic sensor management for multitarget tracking.” in *Proc. 43rd IEEE Conf. Decision and Control*, Paradise Island, Bahamas, Dec. 2004, pp. 722–727.

[157] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.

[158] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[159] D. P. Bertsekas and D. A. Castanon, “Rollout algorithms for stochastic scheduling problems.” *J. Heuristics*, vol. 5, no. 1, Apr. 1999, pp. 89–108.

[160] E. K. P. Chong *et al.*, “A framework for simulation-based network control via hindsight optimization,” *Proc. 39th IEEE Conf. Decision and Control*, Sydney, Australia, Dec. 2000, pp. 1433–1438.

[161] G. Wu *et al.*, “Burst-level congestion control using hindsight optimization,” *IEEE Trans. Autom. Control*, vol. 47, no. 6, 2002, pp. 979–991.

[162] D.P. Bertsekas, *Dynamic Programming and Optimal Control (vol. 2)*, Belmont, MA: Athena Scientific, 2007.

[163] D.P. Bertsekas, “Dynamic programming and suboptimal control: A survey from ADP to MPC,” *European J. on Control: Fundamental Issues on Control*, vol. 11, pp. 4–5, 2005.

[164] Sony FCBEV7500 Camera Block, https://pro.sony.com/bbsc/ssr/cat-camerasindustrial/cat-ciblockcameras/product-FCBEV7500/ accessed Mar. 2017.

[165] C. M. Eaton *et al.*, “Robust UAV Path Planning using POMDP with Limited FOV Sensor,” in *Proc. 2017 IEEE Conf. Control Technology and Application,* Kohala Coast, HI, USA, Aug 2017, pp 1530–1535.

[166] L. W. Krakow and E. K. P. Chong, "Autonomous UAV Control: Balancing Target Tracking and Persistent Surveillance," in *Proc. 2017 IEEE Conf. Control Technology and Application,* Kohala Coast, HI, USA, Aug 2017, pp 1524–1529.

[167] C.M. Eaton *et al.* "Fuel Efficient Moving Target Tracking using POMDP with Limited FOV Sensor," Submitted to *Proc. 2018 IEEE Conf. Control Technology and Application,* Copenhagen, Den, Aug 2018.

[168] BirdsEyeView Aerobatics FireFLY6 Pro, https://www.birdseyeview.aero/ accessed Jan. 2018.

[169] Department of Defense (DoD), "Autonomy Community of Interest (COI) Test and Evaluation, Verification and Validation (TEVV) Working Group, Technology Investment Strategy 2015-2018," Jun 2015.

[170] C. Hagel, "Reagan National Defense Forum Keynote: Simi Valley, CA.," Department of Defense, Nov. 2014.

[171] Department of Defense (DoD), "Defense Science Board Summer Study on Autonomy." Jun. 2016.

[172] National Research Council (NRC), "Autonomy Research for Civil Aviation: Toward a New Era of Flight." Jun. 2014.

[173] American Institute of Aeronautics and Astronautics (AIAA), "Roadmap for Intelligent Systems in Aerospace," Jun. 2016.

[174] United States Air Force (USAF) "Small Unmanned Aircraft Systems (SUAS) Flight Plan: 2016-2036," Apr. 2016.

[175] D. Sprott, L. Wilkes, "Understanding Service-Oriented Architecture," Accessed August 6, 2017. https://msdn.microsoft.com/en-us/library/aa480021.aspx.

[176] D. Kingston, S. Rasmussen, and L. Humphrey, "Automated UAV Tasks for Search and Surveillance," IEEE Multi-Conference on Systems and Control, Sept. 2016.

[177] M. Thompson, "Costly Flight Hours," *Time*, Apr. 2, 2013. http://nation.time.com/2013/04/02/costly-flight-hours/.

[178] D. Kingston, "AFRL Summer of Innovation," Safe and Secure Systems and Software Symposium, Dayton, OH. Aug. 2017.

[179] Air Force Research Laboratory (AFRL), "Open UxAS software repository," GitHub, 2017, https://github.com/orgs/afrl-rq (accessed Sept 3, 2017.)

[180] Institute of Electrical and Electronics Engineers (IEEE), "610.12-1990 - IEEE Standard Glossary of Software Engineering Terminology," Dec. 1990.

[181] ASTM F3269-17, "Standard Practice for Methods to Safely Bound Flight Behavior of Unmanned Aircraft Systems Containing Complex Functions," ASTM International. www.astm.org. (accessed January 3, 2018)

[182] G. Warwick, "LockheedâĂŹs Skunk Works Demos Autonomy for Unmanned Loyal Wingman," *Aerospace Daily*, April 11, 2017. http://aviationweek.com/defense/lockheed-s-skunk-works-demos-autonomy-unmanned-loyal-wingman. (Accessed Dec 23, 2017)

[183] Calspan, "Variable Stability Learjet," http://www.calspan.com/services/aircraft-operation/flight-testing/sensors-airborne-services-test-beds. (accessed Aug 9, 2017)

[184] G. Norris, "NASAâĂŹs Traveler To Demo âĂŸTrustworthyâĂŹ UAS Autonomy," *Aviation Week and Space Technology*, March 28, 2016. http://aviationweek.com/commercial-aviation/nasa-s-traveler-demo-trustworthy-uas-autonomy (accessed Nov 3, 2017).

[185] D. Scheidt, *et al.*, "Safe Testing of Autonomous System Performance," Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC), November 30 - December 4, 2015, Orlando, Florida.

[186] Swift Radioplanes (SRP), "Lynx FarScight," https://srp.aero/lynx/ (accessed Aug 9, 2017).

[187] V. Machi, "Air Force Investing in Live-Virtual-Constructive Technology," *National Defense*, March 3, 2017. http://www.nationaldefensemagazine.org/articles/2017/3/3/air-force-investing-in-live-virtual-constructive-technology (accessed Aug 9, 2017).

[188] Test Resource Management Center (TRMC), "Test and Training Enabling Architecture," https://www.tena-sda.org/display/TENAintro/Home (accessed Sep. 3, 2017)

[189] Naval Air Warfare Center Aircraft Division (NAWC-AD), "Joint Integrated Mission Model," http://www.navair.navy.mil/nawcad/index.cfm?fuseaction=home.content_detail&key=457CD597-AAA6-47A0-82C5-D6A914AF6AC4 (accessed Sep. 3, 2017)

[190] P. Clive, *et al.*, "Advanced Framework for Simulation, Integration and Modeling (AFSIM)," International Conference of Scientific Computing. Paris, France, October 29-30, 2015.

[191] M. Cooper, "AFRL Shares UAS software to further research," *sUAS News*, September 5, 2017. https://www.suasnews.com/2017/09/afrl-shares-uas-software-research/ (accessed Nov 5, 2017)

[192] UAV Factory, "Penguin UAV System," www.uavfactory.com (accessed Sep 3, 2017).

[193] Arcturus UAV, "Arcturus T-20," arcturus-uav.com/product/t-20 (accessed Sep 3, 2017).