MODELING AND SIMULATION OF VEHICLE PERFORMANCE IN A UAV
SWARM USING HORIZON SIMULATION FRAMEWORK

A Thesis

Presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Aerospace Engineering

by

Adam Frye

October 2018

© 2018
Adam Frye
ALL RIGHTS RESERVED

COMMITTEE MEMBERSHIP

TITLE: Modeling and Simulation of Vehicle Performance in a UAV Swarm using Horizon Simulation Framework

AUTHOR: Adam Frye

DATE SUBMITTED: October 2018

COMMITTEE CHAIR: Eric Mehiel, Ph.D.
Professor
Aerospace Department

COMMITTEE MEMBER: Kira Abercromby, Ph.D.
Professor
Aerospace Department

COMMITTEE MEMBER: Paulo Iscold, Ph.D.
Professor
Aerospace Department

COMMITTEE MEMBER: Colleen Kirk, Ph.D.
Professor
Mathematics Department

ABSTRACT

Modeling and Simulation of Vehicle Performance in a UAV Swarm using Horizon Simulation Framework

Adam Frye

A UAV swarm is simulated using Horizon Simulation Framework. The asset utilized for the swarm agent is a simplified model of the MQ-1 Predator, a large fixed-wing aircraft. The simulated swarm utilizes a decentralized cooperative control approach to command the assets through the use of digital pheromones and a pheromone map. Each vehicle operates at steady-state flight conditions of 36 m/s with an altitude of 1,800 m, and utilize an LQR set-point controller to maneuver through the pheromone map. All pheromone and aircraft related models are written in Python to expand the HSF scripting capability and include airborne scenarios. The simulation study focuses in the variation of three parameters in the repelling pheromone model. The first two are the update and deposit parameters with values of 2, 10, and 18. The third is the threshold parameter with values of 1e-02, 1e-10, and 1e-18. The lower parameter values provide more time-on-target while the higher parameters allow the swarm to search the surrounding area by only visiting the grid-space once.

ACKNOWLEDGMENTS

This thesis is dedicated to all “Magnificent Bastards” from 2nd Battalion 4th Marines, and to all of my other 5th Marine Regiment brothers whom I served with. Thanks to Cpl Dustin Libby and SgtMaj Joseph Ellis for motivating and influencing me to never give up on my goals. Semper Fi brothers, you will never be forgotten.

Thank you to Dr Paulo Iscold and Dr Colleen Kirk for taking time out of their busy schedule to be part of my committee. Thank you to Dr Kira Abercromby for making aerospace courses fun and more interesting; thanks for giving me the inspiration to pursue this topic. Thanks to all the faculty and staff of the Aerospace Department for all your passion and dedication to students. Thanks to my family and friends for their support. Thanks to my Hufflepuff friend Karen Kauffman for introducing me to some great reading material, and for all the riveting programming and hackathon conversations. Thanks to my life mentors and friends, Dr Richard Robertson and Craig Snow, for all their valuable advice, and a special thanks to Dr Eric Mehiel for being my advisor. Thank you for being a positive influence in my life and for introducing me to the world of dynamics and control that I have come to enjoy so very much. Thank you for all your guidance and patience with me during the writing of this thesis.

TABLE OF CONTENTS

	Page
LIST OF TABLES	ix
LIST OF FIGURES	x
LIST OF LISTINGS	xv
CHAPTER	
1 Introduction	1
1.1 Modeling and Simulation Analysis	1
1.2 Horizon Simulation Framework	3
1.2.1 System Model	4
1.2.2 Scheduling Algorithm	6
1.2.3 XML Input Files	7
1.2.4 Previous Work in HSF	8
1.3 Swarming	8
1.3.1 Centralized Control	9
1.3.2 Decentralized Cooperative Control	9
1.4 Thesis Statement	10
2 Literature Review	12
2.1 Cost Functions	12
2.2 Artificial Potential Fields	13
2.3 Digital Pheromones	14
2.4 Particle Swarm Optimization	15
2.5 Choosing an Algorithm	16
3 Environmental Models	17
3.1 Atmosphere Model	17
4 Digital Pheromones Model	21
5 UAV Model	26
5.1 MQ-1 Predator	27
5.2 Reference Frames	30
5.3 Aerodynamics	33

5.3.1	Digital DATCOM	36
5.3.2	Athena Vortex Lattice (AVL)	37
5.4	Guidance, Navigation, and Control (GNC)	39
5.4.1	Frenet-Serret Curvature and Torsion	39
5.4.2	Equations of Motion	42
5.4.3	Linear Quadratic Regulator (LQR)	45
5.5	Subsystems	48
5.5.1	Controller	49
5.5.2	Pheromone	52
5.5.3	Motor	52
5.5.4	Camera	54
6	Scenario Simulation	55
6.1	Prepare XML Files	57
7	Simulation Results	59
7.1	Control Case: 1 Asset	60
7.2	Case: 2 Assets	67
7.3	Case: 3 Assets	73
7.4	Case: 4 Assets	78
8	Conclusion and Future Work	85
8.1	Conclusion	85
8.2	Future Work	86
	BIBLIOGRAPHY	88
	APPENDICES	
A	Environmental Model Data	93
A.1	HSF Atmosphere Model	93
A.2	PDAS Atmosphere Model	94
B	UAV Model Data	95
B.1	Digital DATCOM Data	97
B.2	Athena Vortex Lattice Data	102
C	Simulation Parameter Files	109
C.1	Stability and Control Derivatives	110
C.2	Set-Point Control Matrices	111

C.3	Simulation Batch File	113
D	Simulation Output Data	114
D.1	Swarm: 2 Assets	114
D.2	Swarm: 3 Assets	117
D.3	Swarm: 4 Assets	121

LIST OF TABLES

Table	Page
3.1 1976 U.S. Standard Atmosphere reference data for geopotential heights. Separated by temperature lapse curve break points.	18
5.1 MQ-1 Predator general characteristics [4].	29
6.1 Simulation scenario variation parameters. There are three different pheromone threshold values for every deposit, three different deposits for every update value, and three different update values for every swarm size.	56
7.1 Control case output data.	66
7.2 2 asset swarm distance summary data.	72
7.3 3 asset swarm distance summary data.	78
7.4 4 asset swarm distance summary data.	83
A.1 1976 U.S. Standard Atmosphere data generated by HSF for various geometric altitudes.	93
A.2 1976 U.S. Standard Atmosphere data generated by PDAS for various geometric altitudes.	94
B.1 Lift coefficient, C_L , data obtained from the Digital DATCOM and AVL software.	95
B.2 Max roll angle, ϕ_{max} data calculated using the corresponding reference airspeed and a stall speed of 27.2 m/s	95
B.3 MQ-1 Predator model data.	96
C.1 Simulation scenario initial parameters.	109
D.1 2 asset swarm output data.	114
D.2 3 asset swarm output data.	117
D.3 4 asset swarm output data.	121

LIST OF FIGURES

Figure		Page
1.1	Model validation process [30]. Same input parameters passed in for evaluation and model output compared to real system.	2
1.2	Asset subsystem dependencies [38]. Subsystem performance is dependent on the state of other subsystems.	4
1.3	HSF components [28, 38]. Simulation and system parameters are passed into HSF via XML files, the system model contains all asset relevant data including constraints. The main scheduling algorithm generates a list of possible optimized schedules based on the target inputs.	7
2.1	Artificial potential fields [26].	13
2.2	Pheromone propagation [33]. (1) attractive pheromone (2) repelling pheromone (3) propagation of pheromone (4) receding attractive pheromone as UAV climbs in the direction of the gradient.	15
2.3	Particle Swarm Optimization (PSO) [21].	16
3.1	1976 U.S. Standard Atmosphere. HSF standard atmosphere model compared to the PDAS model.	20
3.2	1976 U.S. Standard Atmosphere percent error. HSF standard atmosphere model compared to the PDAS model.	20
4.1	2-dimensional Cartesian grid overlaid across a target terrain.	22
4.2	Pheromone propagation due to neighbors. (a) pheromone propagated in from surrounding neighbors (b) pheromone propagated to middle position from neighbors is only a fraction of the total propagated value.	24
4.3	Pheromone gradient created using data generated by HSF.	25
5.1	Aircraft schematics for the MQ-1 Predator [6]	27
5.2	Earth-centered earth-fixed (ECEF) and north-east-down (NED) reference frames [9].	31
5.3	Body reference frame [12].	32
5.4	Lift coefficient, C_L , curve calculated by DATCOM and AVL for various angles of attack.	34

5.5	Max banking angle for various airspeeds. The square of the max bank angle is proportional to the change in airspeed.	35
5.6	MQ-1 Predator model defined in the DATCOM software.	36
5.7	MQ-1 Predator model defined in the AVL software.	38
5.8	Pre-defined flight trajectories. (a) 0 degree heading change, steady-state flight (b) 45 degree flight move on pheromone map (c) 90 degree heading change.	40
5.9	Command heading adjustment. The black path is the proposed trajectory, the blue arrow represents the current asset path and direction as it deviates from trajectory. The blue dashed line is the concentric based on the asset current location. dx and dy are obtained from the trajectory definition at 180 m , and the overall interval 360 m . They are utilized to calculate μ , and ultimately the sign of Δr to obtain the direction of the heading command angle correction.	42
5.10	Set-point control block diagram [18]. The set-point control input, r , is passed in and multiplied by matrices F and N to obtain the new equilibrium condition points. The difference between the new input equilibrium point, x_{eq} , and the feedback state, x , is multiplied by K , and combined with the new control equilibrium point to define the control input, u . The output vector z contains all controlled output variables.	48
5.11	Controller response to a 45 degree heading curve. (a) predefined curve trajectory, and path traveled by UAV asset (b) trajectory error between UAV path and predefined curve (c) velocity changes throughout maneuver (d) changes in Euler angles throughout maneuver (e) altitude change throughout maneuver.	50
5.12	Controller response to a 90 degree heading curve. (a) predefined curve trajectory, and path traveled by UAV asset (b) trajectory error between UAV path and predefined curve (c) velocity changes throughout maneuver (d) changes in Euler angles throughout maneuver (e) altitude change throughout maneuver.	51
5.13	Rotax 914F four-cylinder engine [32].	53
5.14	Rotax 914F engine performance graphs [32]. (a) engine fuel consumption (b) engine power performance (c) engine torque.	53
5.15	AN/AAS-52 Multi-Spectral Targeting System manufactured by Raytheon [31].	54
6.1	Kilometers per degree of longitude based on the current latitude. The distance per degree is the largest at the equator, and the smallest at the poles.	58

7.1	Control case - lowest parameters, target distance through time. (left) Target distance from start of simulation. (right) Target distance from <i>260 sec</i> to the end of the simulation with the mean distance marked by a dashed red line, and the max distance is marked by a red circle.	60
7.2	Control case - lowest parameters, asset navigation through pheromone map. (left) asset path traveled from start position to target location from start to end of simulation. (right) asset path traveled at vicinity of target location to the end of the simulation.	61
7.3	Control case - lowest parameters, asset fuel consumption and time-on-target.	61
7.4	Control case - lowest parameters, asset dynamic state. (a) body frame axial and lateral velocity (b) body frame normal velocity, and asset altitude (c) Euler angles (d) angle rates.	62
7.5	Control case - highest parameters, target distance through time. (left) Target distance from start of simulation. (right) Target distance from <i>260 sec</i> to the end of the simulation with the mean distance marked by a dashed red line, and the max distance is marked by a red circle.	63
7.6	Control case - highest parameters, asset navigation through pheromone map. (left) Asset path traveled from start position to target location from start to end of simulation. (right) Asset path traveled at vicinity of target location to the end of the simulation.	64
7.7	Control case - highest parameters, asset fuel consumption and time-on-target.	65
7.8	Control case - highest parameters, asset dynamic state. (a) Body frame axial and lateral velocity (b) Body frame normal velocity, and asset altitude (c) Euler angles (d) angle rates.	65
7.9	Case: 2 assets - lowest parameters, target distance through time. Target distance from <i>260 sec</i> to the end of the simulation with the mean distance marked by a dashed red line, and the max distance is marked by a red circle.	68
7.10	Case: 2 assets - lowest parameters, asset navigation through pheromone map. (left) Asset path traveled from start position to target location from start to end of simulation. (right) Asset path traveled at vicinity of target location to the end of the simulation.	68
7.11	Case: 2 assets - lowest parameters, asset dynamic state. (a) Asset 1 Euler angles (b) Asset 2 Euler angles.	69
7.12	Case: 2 assets - lowest parameters, asset time-on-target.	70

7.13	Case: 2 assets - highest parameters, target distance through time. Target distance from <i>260 sec</i> to the end of the simulation with the mean distance marked by a dashed red line, and the max distance is marked by a red circle.	70
7.14	Case 2 assets- highest parameters, asset navigation through pheromone map. (left) Asset path traveled from start position to target location from start to end of simulation. (right) Asset path traveled at vicinity of target location to the end of the simulation.	71
7.15	Case: 2 assets - highest parameters, asset dynamic state. (a) Asset 1 Euler angles (b) Asset 2 Euler angles.	71
7.16	Case: 2 assets - highest parameters, asset time-on-target.	72
7.17	Case: 3 assets - lowest parameters, asset time-on-target.	73
7.18	Case: 3 assets - lowest parameters, target distance through time. Target distance from <i>260 sec</i> to the end of the simulation with the mean distance marked by a dashed red line, and the max distance is marked by a red circle.	74
7.19	Case: 3 assets - lowest parameters, asset navigation through pheromone map. (left) Asset path traveled from start position to target location from start to end of simulation. (right) Asset path traveled at vicinity of target location to the end of the simulation.	74
7.20	Case: 3 assets - lowest parameters, asset dynamic state. (a) Asset 1 Euler angles (b) Asset 2 Euler angles (c) Asset 3 Euler angles. . . .	75
7.21	Case: 3 assets - highest parameters, target distance through time. (left) Target distance from start of simulation. (right) Target distance from <i>260 sec</i> to the end of the simulation with the mean distance marked by a dashed red line, and the max distance is marked by a red circle.	76
7.22	Case 3 assets- highest parameters, asset navigation through pheromone map. (left) Asset path traveled from start position to target location from start to end of simulation. (right) Asset path traveled at vicinity of target location to the end of the simulation.	76
7.23	Case: 3 assets - highest parameters, asset time-on-target.	77
7.24	Case: 3 assets - highest parameters, asset dynamic state. (a) Asset 1 Euler angles (b) Asset 2 Euler angles (c) Asset 3 Euler angles. . .	77
7.25	Case: 4 assets - lowest parameters, asset time-on-target.	79
7.26	Case: 4 assets - lowest parameters, target distance through time. Target distance from <i>260 sec</i> to the end of the simulation with the mean distance marked by a dashed red line, and the max distance is marked by a red circle.	80

7.27	Case: 4 assets - lowest parameters, asset navigation through pheromone map. (left) Asset path traveled from start position to target location from start to end of simulation. (right) Asset path traveled at vicinity of target location to the end of the simulation.	80
7.28	Case: 4 assets - lowest parameters, asset dynamic state	81
7.29	Case: 4 assets - highest parameters, asset time-on-target.	81
7.30	Case: 4 assets - highest parameters, target distance through time. (left) Target distance from start of simulation. (right) Target distance from <i>260 sec</i> to the end of the simulation with the mean distance marked by a dashed red line, and the max distance is marked by a red circle.	82
7.31	Case 4 assets- highest parameters, asset navigation through pheromone map. (left) Asset path traveled from start position to target location from start to end of simulation. (right) Asset path traveled at vicinity of target location to the end of the simulation.	82
7.32	Case: 4 assets - highest parameters, asset dynamic state. (a) Asset 1 Euler angles (b) Asset 2 Euler angles (c) Asset 3 Euler angles (d) Asset 4 Euler angles.	83

LIST OF LISTINGS

6.1	Target deck XML file	55
B.1	Datcom Input File	97
B.2	XML Engine File	99
B.3	XML Propeller File	100
B.4	AVL Input File	102
B.5	MQ-1 Predator Fuselage Coordinates File	106
B.6	Mass Properties File	107
B.7	NACA 2414 Airfoil Coordinates File	108
C.1	Batch File to Run HSF	113

Chapter 1

INTRODUCTION

Swarming is a topic that continuously increases in popularity with the increase in processing power and decrease in cost of electronics. As the trends continue, the capabilities of swarms will also increase with the acquisition of better data, the use of better-quality sensors, the utilization of better reference models, etc. To better understand the aspects of UAV swarming, two scenarios are considered and simulated. The first scenario is a common search and rescue application consisting of locating a missing hiker from a last known location. The swarm travels from an initial point to the designated location and begins expanding its search outward. The second scenario is a reconnaissance operation consisting of monitoring the containment of a brush fire at specified coordinates. For this scenario, the swarm travels from an initial point to the objective and performs a holding pattern to try and maintain as much time-on-target as possible.

To successfully execute the given scenarios, the next two requirements must be met. The first requirement is to define a simulation software that can be utilized to execute swarming scenarios. The next requirement is to define the UAV model that is to be utilized as the base vehicle for all swarm agents in the simulation.

1.1 Modeling and Simulation Analysis

Modeling refers to the creation of a mathematical model of a physical system to obtain results similar to those of the actual physical system. They contain specific attributes and constraints that describe the system as simple and accurately as required for the study [24]. Models are evaluated at discrete steps through a simulation software to

obtain state data. The discrete step parameter can be spatial or temporal but one of the most commonly utilized ones is time. Simulations take time-steps to propagate models through time and calculate the current state of the model at every interval to generate system specific data through a specified period.

Before using a mathematical model for mission specific scenarios, the simulated models must first be validated with data from a known system solution. If the model produces validation data that meets the minimum error tolerances required, then the model is accurate and can then be used with the desired test cases [30]. A visual representation of the validation process for a physical system can be found in fig. 1.1, where both the real system and mathematical model receive the same input parameters, and the output solution from the model is compared to the original system in order to verify if the solution meets the minimum tolerances. Modeling and Simulation (M&S) is used to verify that mission requirements are met in the early design phase. Case studies can ensure better estimates for allocation of resources for the duration of system's life-cycle, reduce the chances of failure to meet system specifications at delivery, and to optimize the performance of the overall system.

Figure 1.1: Model validation process [30]. Same input parameters passed in for evaluation and model output compared to real system.

1.2 Horizon Simulation Framework

Using a simulation to represent a physical system not only ensures all system-level requirements are met throughout the entire mission duration, but also provides system data that can be used for trade-studies and performance analysis. This data is important when defining the Concept of Operations (ConOps) for the overall mission. ConOps provide a guideline of how the system will perform while operating within the bounds of all system requirements and constraints. A well-defined ConOps can be used to accurately predict the types of environments the system will encounter.

Horizon Simulation Framework is a software platform designed to be used for validation and verification (V&V) of system-level requirements. The first iteration of the software was written in C++ in the summer of 2006 by Cal Poly Space Technologies and Applied Research laboratory (CPSTAR) with the support of Cutting Edge Communication, LLC [28, 22]. CPSTAR consisted of a group of engineering students from various disciplines that applied systems engineering principles to space systems. The software aims for modularity, where different environmental models that are part of the HSF library can be used concurrently with each other based on the type of scenario simulated. After various iterations in C++, HSF was redesigned using C# in 2016 to make use of the .NET Framework. The most current version of HSF being used for this thesis is v3.0, and the Integrated Development Environment (IDE) used for editing and compiling the project files is Microsoft Visual Studio 2015. There are two major components that make up HSF. (1) The System Model, and (2) the Scheduling Algorithm.

1.2.1 System Model

The System Model consists of all corresponding subsystems that make up the overall system, described hereafter as the asset. It's a system-of-systems that use model dependencies, constraints, and environmental models to generate the appropriate conditions for the simulation. An example of this is an imaging satellite that contains subsystems like Attitude Determination and Control System (ADCS), Earth observation sensor, solid state data recorder (SSDR), communications (COMM), and power that are dependent on each other. They also provide performance constraints based on the satellite's orbital parameters and specific actions. A depiction of data flow and how the subsystems are dependent on each other can be seen in fig. 1.2.

Figure 1.2: Asset subsystem dependencies [38]. Subsystem performance is dependent on the state of other subsystems.

A performance constraint due to orbital parameters is observed when a satellite in low-Earth orbit (LEO) experiences eclipsing due to Earth's shadow. This affects the

amount of power generated by solar panels over a given orbit which can also affect other subsystems like COMM. A performance constraint induced by a satellite event can be observed when the satellite slews to image a target and changes its attitude, the solar panels may become eclipsed by the satellite and reduce their performance.

In the source code, each subsystem model is required to have various specific functions that allow it to interface with HSF. These consist of dependencies and performance functions. The most important is the *CanPerform* function that is evaluated at every time step. If the subsystem is able to perform its task, then the task is evaluated at the current time interval and the function returns *True*, if not it returns *False* and the task is ignored. An example of the subsystem being unable to perform its task for the current time step could be observed when updating states for an inertial measurement unit (IMU). The IMU gathers data at a specified frequency, if the simulation time interval occurs before the next IMU update, then the IMU subsystem cannot perform the next update and therefore returns false.

Another important function is the *CanExtend* function that is evaluated when a subsystem that is dependent on another subsystem does not have current data to evaluate its task. The function asks the system if the data from the previous time step can be utilized in the current time interval. An example of this occurrence can be observed when a subsystem that updates at every time interval utilizes the IMU states to perform its task. When the current time step falls short of the IMU update interval and the IMU does not update states, the dependent subsystem cannot access the most current data and therefore evaluates the *CanExtend* function to verify that the previous IMU state data remains valid.

1.2.2 Scheduling Algorithm

The scheduling algorithm uses a breadth-first search approach to obtain every possible solution available in the solution space generated by the asset constraints. The algorithm is effective when traversing through data structures connected together, similar to a binary tree. A tree data structure search starts at an initial node and explores all neighboring solution nodes before moving on to explore all the possible solutions related to each of those neighboring nodes. As each new solution is explored and appended to the previous node, the new solution becomes a possible new schedule.

The best example, for physically visualizing the algorithm process, is the imaging satellite containing a list of targets scattered throughout the world. Each target has an imaging priority level which provides a point value system. The objective is to accrue the highest point value based on the targets imaged. Some targets can never be imaged even if they are of the highest priority due to orbit constraints. When a target lies at a much greater latitude than the orbit's inclination, the satellite's path will never cross the target. From the initial point in the simulation, various schedules are generated, where the first consists of imaging the nearest target. Another schedule consists of skipping the first target and imaging the next available target. The schedules vary by skipping a target because the satellite slew-time may prevent the imaging of an incoming high priority target.

As the simulation progresses, the number of possible solution schedules increases exponentially. To mitigate the accrual of too many schedules and reduce the memory use, HSF crops the lowest scoring schedules and retains the others as possible solutions. The number of schedules retained is defined by the user through the simulation Extended Markup Language (XML) input files. A block diagram describing HSF and the flow of data throughout the simulation can be found in fig. 1.3. HSF

receives XML input files to define the simulation. Then the System Model is evaluated at the simulation start time, and based on the asset state and constraints, the Main Scheduling Algorithm evaluates all possible solution schedules.

Figure 1.3: HSF components [28, 38]. Simulation and system parameters are passed into HSF via XML files, the system model contains all asset relevant data including constraints. The main scheduling algorithm generates a list of possible optimized schedules based on the target inputs.

1.2.3 XML Input Files

HSF requires three different XML input files to fully define a simulation scenario [8]. (1) The System Model Parameters, (2) the Target Deck, and (3) the Simulation Parameters. Input file (1) contains all parameters that define asset constraints, subsystems, asset related constants, and whether HSF needs to evaluate any scripted simulation modules via Python files. Input file (2) contains a list and description of all targets to be used by the scheduling algorithm to generate all possible solutions. The last XML input file (3) contains all simulation related parameters such as Julian Date, total simulation time, time-step, and number of desired schedules. All source

code for HSF and case scenarios are hosted through GitHub and are available as a free download.

1.2.4 Previous Work in HSF

There have been two scenarios tested with the current HSF version. The main scenario, which is also used for validation purposes, is the Aeolus test case. The mission consists of imaging as many ground targets as possible out of a 296 target deck with a two-satellite imaging constellation [38]. Each satellite is composed of various subsystems that provide performance data. The subsystems utilized for the Aeolus case are the same subsystems discussed in the System Model subsection.

Another test case is the modeling and simulation of a sounding rocket stabilization system. The rocket contains subsystems that provide data for the control calculations. Code for the controller, state estimator, and Kalman filter were loaded into the control system for a rocket built by Cal Poly Space Systems (CPSS) [23]. The data obtained through flight tests was then used to validate the HSF simulation results.

1.3 Swarming

Swarms are becoming more popular with the increase of processing power, electronic components becoming readily available at lower costs, and the wide range of applications such as formation control, wide area search, and target tracking [11]. Among some favorable swarm attributes are the operational cost savings by requiring fewer personnel to control the UAVs, as well as an extended data gathering range. Choosing the most suitable type of algorithm for a swarm depends on various factors, not limited to the scenario being executed, the types of agents being used for the swarm, as well as the mission range. The two types of control that can be implemented for swarm behavior are centralized and decentralized cooperative control.

1.3.1 Centralized Control

Centralized control relies on a primary source to obtain all relevant tasks and swarm behavior. The primary source, also defined as a leader, obtains data from all other swarm agents, processes the information, and based on the current state of the swarm, the leader assigns tasks to all agents. An effective example of centralized control occurs when a control tower is designated as the leader of a ground vehicle swarm. The leader is able to process all state data updates and assign tasks to all nearby swarm agents.

A similar example for a traveling swarm is different because the leader may not be protected at all times. Even though having a leader agent provides the flexibility of simplifying the other swarm agents, having a leader control the entire swarm is not effective because it introduces a single-point failure and processing time increases as the swarm grows in size [16]. If the leader is taken out during the mission, new tasks cannot be assigned and the swarm may become inoperable. Due to the increase of data being processed as the number of swarm vehicles grow, the leader requires more time to assign tasks to individual agents. The time delay may cause tasks to become invalid.

1.3.2 Decentralized Cooperative Control

Cooperative control allows agents, or subgroups of agents, to process state data as well as data from their immediate surroundings. The agents create individual tasks based on the data processed while maintaining an overall swarm objective. Since there isn't a defined swarm leader, no single-point failure is introduced and therefore this type of control is the most favorable for a UAV swarm. Examples of cooperative control methods include potential fields, digital pheromones, and particle swarm optimization (PSO).

1.4 Thesis Statement

Horizon Simulation Framework (HSF) provides aerospace students a platform for evaluating performance and optimizing task scheduling for different types of scenarios through modeling and simulation of a system. The current HSF iteration tailors the platform to focus on space systems which heavily relies on the scheduling algorithm to optimize tasks because future position and velocity states can be predetermined. The work described throughout this document expands on the software capabilities of HSF to include an airborne system scenario, where a swarm of large fixed-wing unmanned aerial vehicles (UAVs) are controlled through digital pheromones, to search for a missing hiker, or for the swarm to fly a holding pattern at a designated location to assess the containment of a brush fire. A constraint relating to the use of a fixed-wing aircraft is the range of turn direction possibilities. They are limited to a smaller turn scope as compared to a particle swarm element or rotorcraft that may turn in almost any direction on the pheromone map. Since an airborne system's dynamic state cannot be predicted, similar to an orbital trajectory, all tasks are assigned dynamically at every time step throughout the simulation. The swarm behavior and flight performance of each vehicle can be analyzed through the HSF output files after running a simulation scenario.

Pheromone propagation and vehicle performance evaluation is handled through HSF, which is an open-source simulation framework designed for aerospace systems to provide similar analysis results as commercial software. Among the commercial software tools that provide system modeling and simulation analysis capabilities are: Systems Tool Kit (STK), MATLAB Simulink, MapleSim, Wolfram System Modeler, and SimulationX [38]; unfortunately, standard software licenses for commercial use can cost thousands of dollars which can be a cumbersome expense to aerospace students.

The aircraft utilized for the simulation is a generalized version of the MQ-1 Predator, a large fixed-wing aircraft. Vehicle performance data utilized to evaluate the swarm behavior is obtained by varying repelling pheromone parameters. Chapter 2 discusses some of the different swarming methods being utilized by researchers. Chapter 3 describes the 1976 standard atmosphere model utilized to generate air density based on the current altitude. Chapter 4 elaborates on the digital pheromones method. A description of all equations, their significance, and how they are utilized is provided there. Chapter 5 provides a description of the UAV model including relevant subsystems, reference frames, vehicle aerodynamics, and GNC information. Chapter 6 contains all the simulation parameters utilized to define all possible scenarios. Chapter 7 presents and explains the data obtained from HSF, and Chapter 8 provides the conclusion reached based on the simulation data generated as well as the future work that this thesis could contribute to.

Chapter 2

LITERATURE REVIEW

Swarms utilizing decentralized cooperative control focus on individual tasks while working towards an overall mission objective. Since all swarm-agents are built with the same capabilities, there is no defined leader, no single-point failure. Each swarm-agent is able to gather information from its surroundings, process the data, and assign its own task. Some of the more popular swarming algorithms include utilizing cost functions, artificial potential fields, digital pheromones, or particle swarm optimization.

2.1 Cost Functions

Swarming algorithms that utilize cost functions are primarily focused on the optimization aspect of the problem. The cost function is minimized to satisfy the desired solution based on the given parameters and constraints. Cost functions can be seen as energy where the value closest to zero represents the optimal solution, it is the path requiring the least amount of energy. Common applications for this are path planning and collision avoidance [20, 25]. One example consist of using the cost function to optimize the path traveled by a spacecraft while verifying that a safe distance radius is maintained from all other spacecrafts to avoid collisions. Another application for cost functions appears in the form of optimal positioning; the placement of anti-tank land mines at specific locations to cover the widest area with a minimum number of mines [7].

2.2 Artificial Potential Fields

For the artificial potential fields approach, potential functions are utilized to define the guidance law for the swarm-agents. The function describes a gradient-type surface where the objective lies at the lowest point. The swarm-agents start at any given point on the gradient and make their way down to the lowest point by following the direction of the negative gradient. Another way to explain the potential functions is by utilizing point sources and point sinks. The gradient surface can be defined by a single point sink at the objective, or a combination of sources and sinks can be utilized to define terrain. Mountains or adverse terrain can modeled using a point source, the swarm-agents may avoid the terrain by not forming a path through the undesired area. The swarm-agents operate within the space defined by the potential function. Regardless of the swarm-agent's starting location, the swarm-agent travels along the vector field defined by the potential field function to the target location (point sink). Research that utilizes potential fields include formation flight and collision avoidance [26, 10]. The combination of point sinks and sources can create geometric that the swarm-agents can travel to. The result is similar to fig. 2.1, where the objective is a small ring-shaped area. Other applications also include using potential field functions for orbital trajectory control and station keeping although the majority of the work is simulation based, there is limited on-orbit testing utilizing this method [35].

Figure 2.1: Artificial potential fields [26].

2.3 Digital Pheromones

Digital pheromones are designed to resemble pheromones found in nature by attracting or repelling an agent, similar to insect behavior. Pheromones are chemical odors released by many insects to communicate and find each other. Over time, the pheromones propagate and are able to reach other insects that may have been farther away. Digital pheromones are just a mathematical representation of this process. An advantage of using digital pheromones to control a swarm is the minimal information required by the swarm-agent in order to operate. Tasks are self-assigned based on the information acquired from a pheromone map and their immediate surrounding environment. The behavior of the collective is dictated by the pheromone flavor interactions with the swarm-agents. Different pheromone flavors can represent target priorities where part of the swarm, if not all, can quickly change objectives if necessary [34].

An attractive pheromone is used to mark a target location on a pheromone map, and over time the pheromone propagates and creates a gradient for swarm-agents to follow. They climb the positive gradient to reach the target [33]. Collisions among swarm-agents are managed by utilizing a repelling pheromone flavor to mark the swarm-agent's location on the pheromone map; this helps the other swarm-agents know that the specified location is unavailable due to another swarm-agent occupying it. A description and visual representation of the different digital pheromones as well as their propagation can be found in fig. 2.2. Digital pheromone applications focus on airborne swarms or a combination a ground element with an airborne element with an objective of searching or arriving at a specified location. Other applications introduce digital pheromones into space systems scenarios where the pheromones are utilized to mark the path traveled by Earth observing satellites, and other satellites can detect whether an Earth image taken is a duplicate from another satellite [27].

Figure 2.2: Pheromone propagation [33]. (1) attractive pheromone (2) repelling pheromone (3) propagation of pheromone (4) receding attractive pheromone as UAV climbs in the direction of the gradient.

2.4 Particle Swarm Optimization

Particle swarm optimization uses a combination of genetic algorithm methods for evaluating the swarm-agent's position with respect to the target, and digital pheromones for attracting the swarm towards the defined target [19, 21]. The genetic algorithm assigns each swarm-agent a numeric value based on their proximity to the target, higher values are assigned to particles closer to the target. The particle with the highest value and smallest distance to the target becomes the best particle (pBest), and releases an attracting digital pheromone. During the next iteration, the overall swarm moves toward the pBest by following the pheromone gradient. Each agent is once again evaluated, and the pBest from the previous iteration becomes the group's best particle (gBest). An image describing basic PSO can be found in fig. 2.3.

A PSO algorithm provides a reliable method for solving problems where the swarm-agents are not limited by the maneuverability of sudden sharp or wide turns. Applications for this appear to be favorable with robotics where ground swarm-agents can easily adjust their current heading to match the algorithm [39]. In the case of space systems applications, PSO can be utilized for the design phase of a spacecraft.

An example of this can be an Earth orbiting satellite in circular orbit, the PSO algorithm can be utilized to obtain the optimal orbital parameters where the satellite can perform most favorably based on the system level requirements [37].

Figure 2.3: Particle Swarm Optimization (PSO) [21].

2.5 Choosing an Algorithm

The test scenario discussed in this document focuses on a decentralized cooperative control approach which utilizes only digital pheromones to control a UAV swarm. It is an algorithm that is simple to implement in HSF, as well as being unaffected by dynamic changes. Different flavors of attractive pheromone can define different target priority levels. This facilitates future work with HSF where multiple targets are simulated. A higher priority can be assigned to a different target while the swarm is in mid-flight, and a change in the overall swarm behavior can be observed. Due to the massive amounts of data generated by simulations using digital pheromones for every time iteration, validation data is very limited.

Chapter 3

ENVIRONMENTAL MODELS

Environmental models are used to generate the supplementary physical data required for the simulation. They are simple but accurate mathematical representations of physical phenomena encountered by the asset throughout the mission. These can have a significant effect on the overall vehicle performance output by increasing the accuracy of the results, and therefore providing a higher fidelity solution. The complexity used in the creation of each model is dictated by the fidelity requirement imposed by the system-level requirements.

3.1 Atmosphere Model

Since the study consists of the modeling and simulation of a UAV swarm, an atmosphere model is required to simulate the dynamic flight conditions each agent within the swarm encounters throughout the simulation. The atmospheric model utilized for this study is the U.S. Standard Atmosphere from 1976 because of its simplicity and accuracy in computing atmospheric values. As of the writing of this document, the 1976 Standard Atmosphere model is the most recent version released by the U.S. Committee on Extension to the Standard Atmosphere (COESA). For a reader not using the U.S. Standard Atmosphere and trying to create an atmospheric model using the standards outlined by the International Civil Aviation Organization (ICAO), data from this model can be used for validation because the 1976 U.S. Standard Atmosphere yields identical values to the International Standard Atmosphere (ISA) for altitudes of *32 km* and below [5].

The current atmosphere model used in HSF is based on the 1976 U.S. Standard

Atmosphere described in the Public Domain Aeronautical Software (PDAS) website and NASA documents [3, 36]. The atmosphere values are calculated in terms of geopotential height which use Earth’s mean sea level as a reference starting point. The geometric height provided by the asset gets converted to a geopotential height and referenced against an altitude table. Geometric height can be converted to geopotential height by utilizing,

$$h = z \left(\frac{R_E}{z + R_E} \right) \quad (3.1)$$

where h is the geopotential height used with the reference data, z is the geometric height provided by the asset, and R_E is the radius of the Earth defined as the average between semi-major and semi-minor axis at $6,369 \text{ km}$. The altitude table, table 3.1, contains discrete reference values that describe the atmospheric behavior of the model. The maximum geometric altitude that can be calculated is approximately 86 km because there is limited data available for altitudes ranging between 86 km and 150 km . There is not enough data to create an empirical model. Temperature gets calculated using the reference temperature, T_{ref} , the temperature lapse rate, a , that

Table 3.1: 1976 U.S. Standard Atmosphere reference data for geopotential heights. Separated by temperature lapse curve break points.

Altitude (m)	Pressure (Pa)	Temperature (K)	Temperature Lapse Rate (K/m)
0	101,325	288.15	-0.0065
11,000	22,632.1	216.65	0.0
20,000	5,474.89	216.65	0.001
32,000	868.019	228.65	0.0028
47,000	110.906	270.65	0.0
51,000	66.9389	270.65	-0.0028
71,000	3.95642	214.65	-0.002
84,852	0.37338	186.946	0.0

is utilized as the slope for the temperature function, the current geopotential height, h , and the reference altitude, h_{ref} . When the temperature lapse rate is not equivalent to zero, the temperature is calculated with,

$$T(h) = T_{ref} + a(h - h_{ref}) \quad (3.2)$$

where a , and h have the greatest influence on the calculated value. When the temperature lapse rate is zero, the temperature remains constant, $T(h) = T_{ref}$, for all altitudes. Similarly, pressure is calculated using the reference values from table 3.1, current geopotential height, ideal gas constant, R , equivalent to $287.053 \frac{J}{kg \cdot K}$, and the gravity constant, g , equivalent to $9.80665 m/s^2$ as defined in the HSF library. When the temperature lapse rate is not zero, pressure is calculated using the following equation,

$$P(h) = P_{ref} \left(\frac{T(h)}{T_{ref}} \right)^{-g/(aR)} \quad (3.3)$$

however, when the temperature lapse rate is equivalent to zero, the equation changes to,

$$P(h) = P_{ref} \exp \left(\frac{-g(h - h_{ref})}{RT_{ref}} \right) \quad (3.4)$$

where density, $\rho(h)$, with units in kg/m^3 gets calculated using the ideal gas law, where the $P(h)$ is pressure in Pascals (Pa), $T(h)$ is temperature in Kelvin (K), and R is the ideal gas constant. Both pressure and temperature are functions of altitude.

$$\rho(h) = \frac{P(h)}{RT(h)} \quad (3.5)$$

The validating data tables from the PDAS atmospheric model provide rounded output data, and also utilize approximated values for reference variables. HSF utilizes more significant figures for its reference variables and therefore small differences are expected when the generated outputs are compared. The similarities between the HSF and PDAS models can be observed in fig. 3.1. Differences between both models are negligible, and their comparison is provided in fig. 3.2.

Figure 3.1: 1976 U.S. Standard Atmosphere. HSF standard atmosphere model compared to the PDAS model.

Figure 3.2: 1976 U.S. Standard Atmosphere percent error. HSF standard atmosphere model compared to the PDAS model.

Chapter 4

DIGITAL PHEROMONES MODEL

Models that are commonly utilized for airborne systems simulations are written in C# so that other users may have access to them in the future as part of the HSF library. The pheromone model is implemented using Python due to swarming algorithms having very specific attributes that are not generally found in normal aircraft/spacecraft scenarios. Using the scripted capabilities of HSF allows for the creation of almost any type of scenario. All relevant information describing the pheromone map size, target location, pheromone concentrations, etc. are passed in to the simulation through an XML input file, and dynamically stored during the simulation through the Python script model.

The pheromone map model parameters are defined in the environment node section of the XML file because HSF creates one overarching instance that can be accessed by all assets (UAVs) in the simulation. Throughout the scenario, all swarm agents interact with the same pheromone map to obtain relevant position and gradient data regarding their surroundings. When an asset accesses the pheromone map and creates a change, all other assets can perceive the change thereafter. The map geometry being utilized for this model is a Cartesian grid which provides 8 possible directions of movement for an unconstrained particle. Every grid-space on the pheromone map contains a space agent object that keeps track of the different relevant information regarding that particular space. The grid is a 2-D plane digitally overlaid across the target terrain to create the pheromone map. The assets are assumed to travel at an altitude where topographic changes in the land can be disregarded and therefore no terrain obstacles are assumed on the grid. For a representation of the Cartesian grid overlaid across a target area, refer to fig. 4.1. For terrain features that

Figure 4.1: 2-dimensional Cartesian grid overlaid across a target terrain.

rise higher than the pheromone map altitude, an automatic pump with a repelling pheromone can be defined to maintain assets at a safe distance. An automatic pump is utilized to constantly replenish a pheromone concentration at a given location, to keep that pheromone from evaporating.

The implementation of the algorithm requires an attractive pheromone concentration value to be defined on the pheromone map to mark the target position. As time progresses, the pheromone propagates to its neighbors and creates a concentration gradient that the UAVs can follow to traverse toward the predefined target location [33]. The pheromone concentration at any location on the pheromone map is given by,

$$s(\Phi_f, p, t + 1) = E_f \left[(1 - G_f) \left(s(\Phi_f, p, t) + d(\Phi_f, p, t) \right) + g(\Phi_f, p, t) \right] \quad (4.1)$$

where Φ_f represents the specific pheromone flavor being calculated, p is the specified location on the pheromone map, and t is the current time step. $E_f \in (0, 1)$ is the evaporation factor which defines the amount of pheromone remaining after evaporation. Similarly, $G_f \in [0, 1)$ is the propagation factor which defines the amount of pheromone that gets propagated to the neighboring positions on the pheromone map. $s(\Phi_f, p, t)$ is the pheromone concentration at the current location and time-step on the pheromone map. $d(\Phi_f, p, t)$ is the total external pheromone deposit value at the current position. These are deposits that are not accounted for from the previous time step. An example of these are auto-pumps, which continuously supply a pheromone concentration value to the target position to maintain a gradient on the pheromone map. $g(\Phi_f, p, t)$ is the total pheromone being propagated into the current position from the surrounding neighboring positions as observed in fig. 4.2(a). To calculate the pheromone flowing in from all neighboring positions, the following equation is utilized:

$$g(\Phi_f, p, t) = \sum_{p' \in N(p)} \frac{G_f}{|N(p')|} \left(s(\Phi_f, p', t - 1) + d(\Phi_f, p', t - 1) \right) \quad (4.2)$$

where the pheromone value at the current position due to neighbor positions, p' , is obtained through the sum of the pheromone values propagated in from every neighbor element. Figure 4.2(b) shows how the amount of pheromone propagating in from a neighbor is only a fraction of the total propagated value due to propagation to its corresponding neighbors.

Figure 4.2: Pheromone propagation due to neighbors. (a) pheromone propagated in from surrounding neighbors (b) pheromone propagated to middle position from neighbors is only a fraction of the total propagated value.

To avoid collisions among swarm agents while traversing through the gradient, each UAV drops a short-lived repelling pheromone on the location they will be occupying during the next time step. By marking the location of their following move, other UAVs are made aware of the unavailability of that space. When an agent checks the neighboring pheromone concentrations, an occupied space yields a significantly lower concentration than the other spaces. This reduces the probability of selecting an occupied space. To keep track of the different pheromones, various versions of the pheromone map are utilized. One keeps track of all attracting pheromones, and another keeps track of the repelling pheromones. To check for the total pheromone

value within any neighboring location, the following equation is utilized:

$$V(p) = \alpha s(\Phi_l, p) - \beta s(\Phi_v, p) \quad (4.3)$$

where $V(p)$ is the total attractive pheromone value at location p ; Φ_l is the attractive pheromone; Φ_v is the repelling pheromone; and α and β are constants, typically set to a value of 1. Adjusting these constants modifies the system's response to favor one of the pheromone flavors [34]. Running the pheromone model in HSF provides the data utilized to generate fig. 4.3, where the attracting pheromone propagates and creates a gradient across the map.

Figure 4.3: Pheromone gradient created using data generated by HSF.

Chapter 5

UAV MODEL

A UAV model simulates the motion and behavior of an aircraft, and interacts with the environmental models utilized for the scenario to generate dynamic state data specific to the vehicle. The type of system selected for the test case is the large fixed-wing aircraft MQ-1 Predator. Even though this is a very specific aircraft, it should be noted that various generalizing assumptions have been made to obtain all necessary data utilized for running the simulation. Due to the classified and proprietary nature of military aircraft, very little information is publicly available regarding schematics and aerodynamic data. All necessary aircraft dimensions utilized for approximating aerodynamic constants were extrapolated from fig. 5.1. The stability and control derivatives utilized to populate the state transition matrices were obtained using the Athena Vortex Lattice (AVL) and Digital DATCOM software. A description on the use of this software is discussed later in the chapter. A set-point controller is included to maintain steady-state conditions and command a change in heading.

Every UAV asset utilized for the simulation must be defined within the XML input file. The number of assets defined within the file is equivalent to the size of the swarm, where each asset is composed of its corresponding equations of motion and subsystems. They record the asset's current position, velocity, attitude, and subsystem state as it traverses through the pheromone map. The aircraft model accesses the digital pheromones model to choose a direction of movement and designate a local target destination which corresponds to a grid-space location. The asset then maneuvers to the target space by following a predefined curve trajectory, and operating within the mechanical limitations of the vehicle.

Figure 5.1: Aircraft schematics for the MQ-1 Predator [6]

5.1 MQ-1 Predator

The MQ-1 Predator is a UAV manufactured by General Atomics for the U.S. Air Force. It is a medium-altitude, long-endurance, large fixed-wing aircraft initially created as a reconnaissance vehicle under the Mission Design Series (MDS) designation RQ-1 Predator. Where R stands for air reconnaissance, Q refers to an unmanned aerial system, and the 1 is the vehicle series design number [14]. The first deployment of the system occurred in 1995 during the Bosnian conflict in support of NATO while operating from Gjader, Albania [13]. Later, the aircraft was modified to increase the payload capacity and add the capability of carrying up to two AGM-114 Hellfire missiles. This new capability provided operators with the ability to strike a target remotely from Creech Air Force Base in Nevada. The first Predator strike occurred on October 7, 2001 in Afghanistan at the start of Operation Enduring Freedom (OEF); less than a month after the events following the September 11 attacks. In 2002, the designation was officially changed to MQ-1 Predator, where the M refers to multi-mission [4]. Although the aircraft was retired from service on March 9, 2018, other

variants of the aircraft remain in service and are operated by different countries [1].

The UAV model is defined by the various subsystems that make up the aircraft, and by the equations of motion. The subsystems modeled for the current scenario are: the pheromone subsystem that interacts with the pheromone model to select a new direction of flight, and also propagates pheromone concentrations. A controller subsystem that contains all control gain matrices, obtained by previously solving the Algebraic Riccati Equation, utilized with the state transition matrices to maintain level flight. A motor subsystem that models the Rotax 914F engine carried by the MQ-1 Predator, and utilized to calculate the quantity of fuel spent throughout the mission. The last subsystem model included for the simulation is the camera, the range covered by the field-of-view angle has been modified to be equivalent to the size the defined grid-space on the pheromone map. This allows the asset to search their corresponding location in its entirety before traversing to a new position. These subsystems are discussed in more detail within the subsystems section, section 5.5.

One of the generalizing assumptions for the UAV model is the airfoil type. The MQ-1 Predator airfoils are General Atomics' proprietary designs, and therefore not available for public use. The model utilizes a NACA 2414 due to its low camber, moderate thickness, and possible use in glider applications. Data coordinates for this airfoil can be found in Appendix B. The wings are assumed to be rigid, they possess no dihedral angle, and it is also assumed that there is no induced dihedral angle throughout flight. Other essential and relevant information utilized for the creation of the subsystems that define this model can be found in table 5.1, the vehicle's general characteristics table.

Since the simulation case is a reconnaissance related mission, the weapons payload mass is ignored, and a total mass of 800 kg is utilized for the asset to represent the vehicle and fuel mass. An aircraft velocity of 36 m/s , approximately 70 knots , is

utilized for steady-state flight conditions to conserve fuel throughout the mission. Based on the defined cruise speed, the corresponding altitude is $1,800\text{ m}$. At these conditions, the estimated air density is equivalent to 1.027 kg/m^3 with a local speed of sound of 333 m/s . Steady state flight conditions for this model are assumed to occur at a zero angle of attack. The stability and control derivatives are also assumed to remain constant throughout the duration flight regardless of the change in angle of attack during maneuvering events. Having the stability and control derivatives remain constant results in constant state transition matrices. Stability and control derivatives, and state transition matrices are discussed in more detail later in this chapter under section 5.4 Guidance, Navigation, and Control.

Table 5.1: MQ-1 Predator general characteristics [4].

Component	Description
Powerplant	Rotax 914F four-cylinder engine
Max Thrust	115 horsepower
Wingspan	55 feet (16.8 meters)
Fuselage Length	27 feet (8.22 meters)
Height	7 feet (2.1 meters)
Dry Weight	1,130 pounds (513 kilograms)
Max Takeoff Weight	2,250 pounds (1,020 kilogram)
Fuel Capacity	665 pounds (100 gallons, 300 kg)
Payload	450 pounds (204 kilograms)
Cruise Speed	approximately 84 mph (70 knots) to 135 mph
Range	770 miles (675 nautical miles, 1,250 km)
Max Altitude	25,000 feet (7,620 meters)

5.2 Reference Frames

Throughout the simulation, different reference frames are utilized depending on the type of information being calculated or how it is being processed. One reference frame is the Earth-Centered Inertial (ECI) frame, where the x and y axes are located on the equatorial plane. The x-axis extends outward from the center of the Earth in the direction of the vernal equinox, also known as the First Point of Aries. The z-axis is perpendicular to the equatorial plane and extends outward through the North Pole and the Earth's axis of rotation. The y-axis is then defined by the cross product between the x and z axes.

Similar to the ECI frame is the Earth-Centered, Earth-Fixed (ECEF) frame, where the x and y axes are also located on the equatorial plane. The x-axis extends outward from the center of the Earth and through the prime meridian. This refers to the zero degree longitude line that passes through Greenwich in London. The z-axis is defined as being perpendicular to the equatorial plane and extends out through the North Pole. The reference frame is completed by defining the y-axis as the cross product between the x and z axes. The primary difference between the ECI and ECEF frame is that the ECEF frame rotates with the Earth along its axis. Since the planet rotation and the current position of the planet in space are not relevant for this particular case, a non-rotating frame is considered, and both Earth-Centered Inertial (ECI) and ECEF frame become the same. All references to the inertial frame henceforth refer to the ECEF frame. The inertial reference frame is utilized to obtain the distance between the target and starting swarm position, and it is discussed later in chapter 6.

The mission defined in the test case occurs within a short range relative to the size of the Earth. For convenience of tracking distances traveled, a flat Earth assumption is utilized and a North, East, Down (NED) reference frame is defined, typically at the starting location of the vehicle. In a flat Earth, the ground is assumed to possess

no Earth curvature with the N-axis pointing north along the longitudinal line. The E-axis points east along the latitude line. The D-axis points downward toward the center of the Earth; an altitude, h , to a location in the sky would have a negative value. For better clarification on the reference frames, a visual representation of the ECEF and NED frames can be found in fig. 5.2.

Figure 5.2: Earth-centered earth-fixed (ECEF) and north-east-down (NED) reference frames [9].

Another reference frame that is utilized throughout the simulation is the body frame. As the asset traverses through the pheromone map, the data generated through the equations of motion is calculated in the reference frame of the asset. The origin of the body frame is generally defined at the asset's center of mass due to simplify the calculation of the dynamic states due to vehicle rotation. The x-axis extends out from the origin and through the nose of the aircraft. The z-axis points down towards the bottom of the aircraft. The y-axis completes the frame and extends out perpendicular to the x and z axes and in the direction of the right wing. The UAV model is allowed to have a 6 degree-of-freedom motion by including roll, pitch, and yaw. A positive roll is defined as the moment occurring about the x-axis by lowering

the right wing. A positive pitch is defined as the moment generated about the y-axis by raising the nose of the aircraft. A positive yaw is defined as the moment about the z-axis when the nose of the aircraft moves in the direction of the positive y-axis. A diagram defining the body frame and directions of positive moments is located in fig. 5.3.

Figure 5.3: Body reference frame [12].

The majority of the calculations occur in the body frame but some data is easier to visualize and understand in a different frame, like the NED frame. Depending the type of scenario being tested, some environmental models may also provide information in a different frame; therefore, conversions between reference frames is necessary. A direction cosine matrix (DCM) allows for the rotation of different frames. The most commonly utilized conversion throughout the simulation occurs between the body and NED frame. The following DCM, $\mathbf{C}_{NED}^{body}$, is utilized to convert data from NED to body.

$$\begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix} \quad (5.1)$$

$$\mathbf{x}_{body} = \mathbf{C}_{NED}^{body} \mathbf{x}_{NED} \quad (5.2)$$

5.3 Aerodynamics

Stability and control coefficients for an aircraft are generally obtained through wind tunnel testing or from estimates based on the vehicle geometry. During the design phase, reference documentation, such as USAF DATCOM, along special software, like Athena Vortex Lattice (AVL) and Digital DATCOM, can be utilized to approximate these values and provide an estimated vehicle flight performance. Both software applications are very efficient at estimating the corresponding coefficients but they have their own limitations. Digital DATCOM does not include accurate estimates of coefficients for lateral movement, and some stability and control derivatives remain constant regardless of change in angle of attack. However, the estimated stability and control coefficients exhibit realistic trends because Digital DATCOM takes into consideration flow separation as the angle of attack increases. Digital DATCOM estimates are based on empirical data correlation and best estimates of aerodynamics. AVL provides C_L trends with stability and control derivatives as the angle of attack changes, however, the same C_L trends remain constant even when the angle of attack is high enough to cause flow separation and a drastic change in the estimated coefficients. Since the values generated by both programs are similar, the more realistic estimations have been utilized for this scenario.

Before the first simulation is executed, a few calculations need to occur to find the mechanical limitations of the aircraft, and to introduce a simulation constraint. Since the asset travels on a 2D grid, and it remains at a quasi-constant altitude, the only constraints being considered for this case are the roll angle, and angle of attack limitations. To find the maximum banking angle before stalling, the asset's

stall speed must first be calculated using the following equation,

$$V_{stall} = \sqrt{\frac{2mg}{\rho S C_{L_{max}}}} \quad (5.3)$$

where m is the asset's mass at 800 kg , and g is the gravity constant at 9.80665 m/s^2 . ρ is the local air density, 1.027 kg/m^3 , at $1,800 \text{ m}$, S is the wing reference area at 12.84 m^2 , and $C_{L_{max}}$ is the maximum lift coefficient generated by the asset which can be obtained by looking at fig. 5.4 or the data table provided in Appendix B.

Figure 5.4: Lift coefficient, C_L , curve calculated by DATCOM and AVL for various angles of attack.

For the max lift coefficient case, the DATCOM approximation provides the more realistic data because the values decrease after reaching a critical angle of attack, while the values obtained from AVL continue to increase linearly. The maximum coefficient approximation occurs when the asset achieves an angle of attack of *16 degrees*, yielding an equivalent lift coefficient of *1.61*. The stall speed calculated based on the given conditions is *27.2 m/s* with a max banking angle of approximately *55 degrees* for a

steady-state flight airspeed of 36 m/s . As long as the asset operates above the stall speed, below the max angle of attack, and below the max banking angle, the vehicle will continue to perform nominally throughout the simulation.

$$\phi_{max} = \arccos \left(\left(\frac{V_{stall}}{V} \right)^2 \right) \quad (5.4)$$

The max banking angle equation is a function of airspeed. Ideally, the max bank angle previously calculated should suffice for every scenario; however, while the asset traverses to different locations on the pheromone map, the total airspeed varies depending on the maneuver and current attitude. The square of the max banking angle is proportional to the airspeed. Assuming that an airspeed decrease of up to 10% is expected throughout the simulation, the new max bank angle becomes 45 degrees based on an airspeed of 32.4 m/s . A graph showing the different max roll angles for various airspeeds can be found in fig. 5.5. The airspeed utilized ranges from the stall speed of 27.2 m/s to 40 m/s .

Figure 5.5: Max banking angle for various airspeeds. The square of the max bank angle is proportional to the change in airspeed.

5.3.1 Digital DATCOM

The USAF Stability and Control DATCOM (Data Compendium) is a reference document created by the U.S. Air Force to predict aerodynamic coefficients for various vehicle geometries. It was compiled during the period of September 1975 to September 1977 using empirical data correlations and recordings of best knowledge in aerodynamics. A FORTRAN software that contained the calculation methods of the DATCOM document was released in 1978, along with the complete three volumes that make up the full DATCOM reference, to facilitate the calculation process [17]. The aerodynamic coefficient approximations are useful during the design phase because they provide a medium for calculating aircraft behavior and physical limits. Unfortunately, the Digital DATCOM software does not provide accurate calculations for coefficients related to lateral motion. The software was also unable to calculate any coefficients for the inverted V-tail as part of the fuselage. The software also had some trouble rendering the inverted v-tail and vertical fin; therefore the tail in fig. 5.6 appears to be bigger than in the AVL model even though they utilize the same dimensions.

Figure 5.6: MQ-1 Predator model defined in the DATCOM software.

5.3.2 Athena Vortex Lattice (AVL)

The AVL software was written by Harold Youngren in 1988, based on the NASA software using the Vortex Lattice Method (VLM), as part of the MIT Athena TODOR software collection for aeronautics and astronautics students. TODOR is composed of simulation and analysis programs that allow students to model a myriad of scenarios from flow over airfoils to orbital decay without the use of wind-tunnel testing or help from NASA. There have been various modifications to the AVL software since inception. The changes were incorporated by Mark Drela and Harold Youngren [15].

AVL assumes rigid structures when applying the extended vortex lattice model to the user-defined object. Any lift-induced dihedral angle occurring due to bending of the wings is not accounted in the approximations. The software is also limited to calculations with small angles of attack and small sideslip angles. Approximations are not accurate at high angles because the aerodynamic effects cannot be properly modeled with the vortex lattice model. An example of this was previously observed in fig. 5.4 where the C_L in AVL continued to increase even after the critical angle of attack was reached. Slender-body model calculations are utilized to approximate the data relating to the fuselage. The modeled body is divided into various sections depending on how quickly the cross-sectional area changes. The number of sections can be defined in the input file.

To define a the aircraft schematics in AVL, a start reference point is required. The AVL file created for the simulation contain a stating reference point at the nose of the vehicle. Similar to the body frame, the x-axis is situated at the centerline of aircraft. The difference is that the positive x-axis points in the aft direction. The z and y axes are defined the same way as the body frame. Lifting bodies, like the wings, are defined as one rigid piece. Depending on the shape of the wing, and control surfaces, it is divided into sections. The tip of the wing is defined first, the next section is

defined when there is a significant change. For this case, it is the start of the aileron.

The next section is defined when the control surface ends, and the one after is defined at the axis of symmetry since the chord length at the root is different from the chord length at the tip. The same process occurs from the axis of symmetry to the tip of the wing so that it is defined as one rigid part. The tail is defined with a similar process. For a geometry model generated by AVL, see fig. 5.7. Even though the control surfaces are not displayed on the vehicle geometry, the coefficient approximation outputs do account for them. The input file utilized to generate the image can be found in Appendix B. AVL is also able to provide the stability and control derivatives for the coupled equations of motion which are used to populate the state transition matrix. Since the software is able to account for both axial and lateral motion, the AVL stability and control derivatives can be directly utilized with the simulation.

Figure 5.7: MQ-1 Predator model defined in the AVL software.

5.4 Guidance, Navigation, and Control (GNC)

One of the most important parts of the UAV model is the GNC section. This is how the asset decides on the type of trajectory to take to reach the next destination, how it can maintain the steady-state flight conditions, and how quickly the aircraft should respond to a command. To generalize movement between grid-spaces, some general trajectories have been defined for the asset to follow while traversing through the pheromone map. The movements are constrained to the five grid-spaces separated by *45 degrees* starting from a *90 degree* turn, and ending in a *-90 degree* turn. Even though the assets can move in *45 degree* intervals, the asset attitude can only change in *90 degree* intervals when a full turn is performed. The *90 degree* turn changes the asset attitude by *90 degrees*, but the *45 degree* turn maintains the initial attitude due to the defined trajectory. Having the asset attitude change at a greater interval helps maintain movement consistency by utilizing the same defined trajectories with all upcoming maneuvers.

5.4.1 Frenet-Serret Curvature and Torsion

The Frenet-Serret formulae define a reference frame similar to the body frame which is beneficial when a curved trajectory has been previously defined. The main curved trajectories consist of a *45*, and *90 degree* turn on the pheromone map. The specified turn does not directly represent the maneuvers performed by the asset, fig. 5.8 shows the trajectories defined to achieve each turn. The *45 degree* turn does not change the final asset attitude, it only moves the asset in the described direction. An attitude of *45* does not work well with the predefined trajectories on the pheromone map; for this reason, the asset attitude can only change in *90 degree* intervals. A mirrored trajectory is utilized when the asset is required to turn in the opposite direction. The Frenet-Serret formulae create a reference frame that calculates the curvature

and torsion of a curve as a point-mass traverses through the defined path. The three vectors that define the Frenet-Serret frame are the tangent unit vector, $\mathbf{T}$, the normal unit vector, $\mathbf{N}$, and the binormal unit vector, $\mathbf{B}$. This frame is also known as the TNB frame [29] and can be observed in fig. 5.8 as being defined by the turn trajectories. Since the trajectory is defined as a 2D curve, only the $\mathbf{T}$ and $\mathbf{B}$ vectors are shown in the image. The $\mathbf{T}$ vector is parallel to the asset's velocity vector, and the $\mathbf{B}$ vector

Figure 5.8: Pre-defined flight trajectories. (a) 0 degree heading change, steady-state flight (b) 45 degree flight move on pheromone map (c) 90 degree heading change.

is parallel to the centripetal acceleration vector. The TNB frame is beneficial with calculating the attitude change required to follow the trajectory path.

To calculate the tangent, normal, and binormal unit vectors, the following formulae are utilized,

$$\mathbf{T}(s) = \frac{\dot{\mathbf{r}}(t)}{|\dot{\mathbf{r}}(t)|} = \frac{\frac{d\mathbf{r}}{dt}}{\frac{ds}{dt}} = \frac{d\mathbf{r}}{ds} = \mathbf{r}'(s) \quad (5.5)$$

$$\mathbf{N}(s) = \frac{\mathbf{r}''(s)}{|\mathbf{r}''(s)|} = \frac{T'(s)}{|T'(s)|} \quad (5.6)$$

$$\mathbf{B}(s) = \mathbf{T}(s) \times \mathbf{N}(s) \quad (5.7)$$

and the differential equations for the TNB frame can be expressed in matrix form the following way,

$$\begin{bmatrix} \mathbf{T}' \\ \mathbf{N}' \\ \mathbf{B}' \end{bmatrix} = \begin{bmatrix} 0 & \kappa & 0 \\ -\kappa & 0 & \tau \\ 0 & -\tau & 0 \end{bmatrix} \begin{bmatrix} \mathbf{T} \\ \mathbf{N} \\ \mathbf{B} \end{bmatrix} \quad (5.8)$$

where κ is the curvature of the trajectory defined as $|\mathbf{r}''(s)|$, and τ is the torsion defined as $(-\mathbf{N} \cdot \mathbf{B}')$ or $\frac{(\mathbf{r}' \times \mathbf{r}'') \cdot \mathbf{r}'''}{|\mathbf{r}'' \cdot \mathbf{r}''|}$. If the asset deviates from the trajectory path due to the controller response, the trajectory error may not be reduced to zero if an error correction is not implemented. When the asset deviates from the trajectory path, it is assumed that the current asset location falls on a curve concentric to the defined trajectory. The difference in the concentric curve radius created by the asset's location compared to the defined trajectory curve radius, Δr , provides a direction to compensate due to offset. By utilizing the interval distance between grid points on the pheromone map, and the trajectory path definition, a dx and dy component can be calculated and translated to an angle. A visual representation of the error correction process is depicted in fig. 5.9. The angle, μ , calculated from the dx and

dy components of the current asset location, is equivalent to the angle created by current trajectory position.

Figure 5.9: Command heading adjustment. The black path is the proposed trajectory, the blue arrow represents the current asset path and direction as it deviates from trajectory. The blue dashed line is the concentric based on the asset current location. dx and dy are obtained from the trajectory definition at 180 m , and the overall interval 360 m . They are utilized to calculate μ , and ultimately the sign of Δr to obtain the direction of the heading command angle correction.

5.4.2 Equations of Motion

The equations of motion (EOMS) describe the behavior of the asset throughout flight in a linear time-invariant system (LTI). The EOMS are defined in the body frame as a 6 degree-of-freedom model where u, v, w represent the axial velocity, the lateral velocity, and the velocity normal to the plane created by u and v respectively.

ϕ, θ, ψ represent the roll, pitch, and yaw angles respectively. These angles are also known Euler angles, and are utilized with a DCM to change reference frames. The DCM provides way to rotate the current frame to the initial reference frame that is generally aligned with NED. The Euler rates for the asset are also calculated with the EOMS and are represented by p, q, r . The force components are designated by X, Y, Z , gravity components are represented by g_x, g_y, g_z , the moments of inertia are $I_{xx}, I_{yy}, I_{zz}, I_{xz}$, and moments generated by rolling, pitching, and yawing motion are defined as L, M, N respectively [12].

$$\begin{aligned}
\frac{du}{dt} &= \frac{X}{m} + g_x + rv - qw \\
\frac{dv}{dt} &= \frac{Y}{m} + g_y - ru + pw \\
\frac{dw}{dt} &= \frac{Z}{m} + g_z + qu - pv \\
\frac{dp}{dt} &= \frac{I_{zz}L + I_{xz}N - I_{xz}(I_{yy} - I_{xx} - I_{zz})p + [I_{xz}^2 + I_{zz}(I_{zz} - I_{yy})]rq}{I_{xx}I_{zz} - I_{xz}^2} \\
\frac{dq}{dt} &= \frac{M - (I_{xx} - I_{zz})pr - I_{xz}(p^2 - r^2)}{I_{yy}} \\
\frac{dr}{dt} &= \frac{I_{xz}L + I_{xx}N - I_{xz}(I_{yy} - I_{xx} - I_{zz})r + [I_{xz}^2 + I_{xx}(I_{xx} - I_{yy})]pq}{I_{xx}I_{zz} - I_{xz}^2} \\
\frac{d\phi}{dt} &= p + (q \sin \phi + r \cos \phi) \tan \theta \\
\frac{d\theta}{dt} &= q \cos \phi - r \sin \phi \\
\frac{d\psi}{dt} &= (q \sin \phi + r \cos \phi) \sec \theta
\end{aligned} \tag{5.9}$$

The EOMS can also be decoupled into longitudinal and lateral-directional equations, and implemented in a state space form of a linear time invariant system for simplified numerical computations using a linear system of equations in the following form:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{5.10}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \tag{5.11}$$

where $\mathbf{x}(t)$ is the state vector containing the state variables as a $[m \times 1]$ column vector; $\mathbf{u}(t)$ is the input vector composed of the control input variables as a $[n \times 1]$ column

vector; and $\mathbf{y}(t)$ is the output vector with all the variables desired from the system, represented as a $[m \times 1]$ column vector. $\mathbf{A}$ is the $[m \times m]$ state matrix that contains the aerodynamic stability derivatives, $\mathbf{B}$ is the $[m \times n]$ input matrix that contains the aerodynamic control derivatives. $\mathbf{C}$ is the $[m \times m]$ output matrix which is composed of the identity matrix, $\mathbf{I}$, for this model, and $\mathbf{D}$ is assumed to be the $[m \times n]$ zero matrix, $\mathbf{0}$. Before arriving at the final definition of eq. (5.10), the previous step for converting the EOMS into state space form is,

$$\mathbf{M}\dot{\mathbf{x}}(t) = \mathbf{A}'\mathbf{x}(t) + \mathbf{B}'\mathbf{u}(t) \quad (5.12)$$

$$\mathbf{A} = \mathbf{M}^{-1}\mathbf{A}' \quad \mathbf{B} = \mathbf{M}^{-1}\mathbf{B}' \quad (5.13)$$

and the longitudinal equations can be calculated by the following:

$$\mathbf{x}^T(t) = [u, w, q, \theta] \quad \mathbf{u}^T(t) = [\eta, \tau] \quad (5.14)$$

$$\mathbf{M} = \begin{bmatrix} m & -\frac{\partial X}{\partial \dot{w}} & 0 & 0 \\ 0 & (m - \frac{\partial Z}{\partial \dot{w}}) & 0 & 0 \\ 0 & -\frac{\partial M}{\partial \dot{w}} & I_{yy} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.15)$$

$$\mathbf{A}' = \begin{bmatrix} \frac{\partial X}{\partial u} & \frac{\partial X}{\partial w} & (\frac{\partial X}{\partial q} - mW_e) & -mg \cos \theta_e \\ \frac{\partial Z}{\partial u} & \frac{\partial Z}{\partial w} & (\frac{\partial Z}{\partial q} + mU_e) & -mg \sin \theta_e \\ \frac{\partial M}{\partial u} & \frac{\partial M}{\partial w} & \frac{\partial M}{\partial q} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \mathbf{B}' = \begin{bmatrix} \frac{\partial X}{\partial \eta} & \frac{\partial X}{\partial \tau} \\ \frac{\partial Z}{\partial \eta} & \frac{\partial Z}{\partial \tau} \\ \frac{\partial M}{\partial \eta} & \frac{\partial M}{\partial \tau} \\ 0 & 0 \end{bmatrix} \quad (5.16)$$

where η is the elevator angle deflection in radians; and τ is thrust in Newtons. U_e and W_e are the steady-state components of velocity, and θ_e is steady-state pitch angle in radians. Similarly, the lateral-directional equations are calculated using the following:

$$\mathbf{x}^T(t) = [v, p, r, \phi, \psi] \quad \mathbf{u}^T(t) = [\xi, \zeta] \quad (5.17)$$

$$\mathbf{M} = \begin{bmatrix} m & 0 & 0 & 0 & 0 \\ 0 & I_{xx} & -I_{xz} & 0 & 0 \\ 0 & -I_{xz} & -I_{zz} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.18)$$

$$\mathbf{A}' = \begin{bmatrix} \frac{\partial Y}{\partial v} & (\frac{\partial Y}{\partial p} + mW_e) & (\frac{\partial Y}{\partial r} - mU_e) & mg \cos \theta_e & mg \sin \theta_e \\ \frac{\partial L}{\partial v} & \frac{\partial L}{\partial p} & \frac{\partial L}{\partial r} & 0 & 0 \\ \frac{\partial N}{\partial v} & \frac{\partial N}{\partial p} & \frac{\partial N}{\partial r} & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (5.19)$$

$$\mathbf{B}' = \begin{bmatrix} \frac{\partial Y}{\partial \xi} & \frac{\partial Y}{\partial \zeta} \\ \frac{\partial L}{\partial \xi} & \frac{\partial L}{\partial \zeta} \\ \frac{\partial N}{\partial \xi} & \frac{\partial N}{\partial \zeta} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

where ξ is the aileron angle deflection in radians; and ζ represents the rudder deflection in radians. The process of deriving the state space form of the EOMS is described in detail in [12]. Fortunately, AVL is able to generate the $\mathbf{A}$ and $\mathbf{B}$ matrices of a specified angle of attack for the coupled EOMS. For the simulated scenario, only the zero angle of attack stability and control matrices are utilized.

5.4.3 Linear Quadratic Regulator (LQR)

A controller is required to maintain the asset's dynamic state at equilibrium conditions. For this model, the steady-state flight is defined as the equilibrium conditions to maintain throughout the simulation duration. To achieve this, a negative full-state

feedback scheme is implemented with a control law of

$$\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t) \quad (5.20)$$

where $\mathbf{K}$ is the $[n \times m]$ control gains matrix. The k-gains are obtained by placing poles on the left-hand side of the plane, the negative real side of the plane provides stability to the system. To find the best pole placement, and values for $\mathbf{K}$, the LQR technique provides a process for calculating the optimal gains matrix by utilizing the control law that minimizes the cost function,

$$J_{LQR} = \int_0^\infty \left[\mathbf{z}(t)^T \mathbf{Q} \mathbf{z}(t) + \rho \mathbf{u}(t)^T \mathbf{R} \mathbf{u}(t) \right] dt \quad (5.21)$$

where $\mathbf{z}(t)^T \mathbf{Q} \mathbf{z}(t)$ can be considered the energy from the control output, and $\mathbf{u}(t)^T \mathbf{R} \mathbf{u}(t)$ the energy from the control input. ρ is a constant that establishes a trade between both energies. $\mathbf{z}(t)$ is the control output, it's a $[l \times 1]$ column vector containing the variables that have been minimized to equilibrium conditions, and take on the state space form of,

$$\mathbf{z}(t) = \mathbf{G}\mathbf{x}(t) + \mathbf{H}\mathbf{u}(t) \quad (5.22)$$

where G and H are utilized to calculate matrices Q and R . To command the asset to maneuver across the pheromone map, a set-point control is required. This allows for certain equilibrium states to vary to a specified value. The number of states that can be commanded depend on the number of control inputs. Therefore, $\mathbf{G}$ becomes a $[l \times m]$ matrix, and $\mathbf{H}$ becomes a $[l \times n]$ square matrix. If the number of commanded states exceeds the number of control inputs, the system may become underactuated and not perform as desired.

Matrices $\mathbf{Q}$ and $\mathbf{R}$ are utilized to solve for $\mathbf{K}$ using the Algebraic Riccati Equation (ARE). They are square matrices with values along the diagonal that can be populated by utilizing Bryson's Rule. The process can be implemented to calculate the values

along the diagonals by using the following,

$$\mathbf{Q}_{ii} = \frac{1}{\max \text{ acceptable value of } \mathbf{z}(t)^2} \quad i \in \{1, 2, \dots, l\} \quad (5.23)$$

$$\mathbf{R}_{jj} = \frac{1}{\max \text{ acceptable value of } \mathbf{u}(t)^2} \quad j \in \{1, 2, \dots, n\} \quad (5.24)$$

and the ARE can then be evaluated for the matrix $\mathbf{P}$ by utilizing one of the ARE equation versions for an LTI system.

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} + \mathbf{G}^T \mathbf{Q} \mathbf{G} - (\mathbf{P} \mathbf{B} + \mathbf{G}^T \mathbf{Q} \mathbf{H})(\mathbf{H}^T \mathbf{Q} \mathbf{H} + \rho \mathbf{R})^{-1}(\mathbf{B}^T \mathbf{P} + \mathbf{H}^T \mathbf{Q} \mathbf{G}) = \mathbf{0} \quad (5.25)$$

For a detailed derivation of the ARE and the creation of the $\mathbf{Q}$ and $\mathbf{R}$ matrices, refer to [18]. The solution obtained by evaluating the ARE is then utilized to calculate the k-gains for the $\mathbf{K}$ matrix.

$$\mathbf{K} = (\mathbf{H}^T \mathbf{Q} \mathbf{H} + \rho \mathbf{R})^{-1}(\mathbf{B}^T \mathbf{P} + \mathbf{H}^T \mathbf{Q} \mathbf{G}) \quad (5.26)$$

To implement set-point control, the control input value, $\mathbf{u}_r$, is passed in and multiplied by matrices $\mathbf{F}$ and $\mathbf{N}$. The set-point then becomes the new equilibrium conditions with an equilibrium input vector, $\mathbf{x}_{eq}$, and an equilibrium control vector, $\mathbf{u}_{eq}$. The closed loop form equation for the set-point control becomes,

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B} \mathbf{K}) \mathbf{x} + \mathbf{B}(\mathbf{K} \mathbf{F} + \mathbf{N}) \mathbf{u}_r \quad (5.27)$$

and a block diagram describing set-point control is shown in fig. 5.10. Where r is the same as the set-point control input, $\mathbf{u}_r$.

Fortunately, MATLAB can facilitate the LQR calculations by solving the ARE, given that the required matrices are passed into the function. The set-point control for the simulated scenario is defined to command the asset heading and altitude. The majority of the command inputs relate only to heading control because the pheromone map is created as a 2D grid. The asset is set to only change altitude in the event that there are no available neighboring grid spaces. The asset is forced to increase altitude, and occupy a grid space that is currently occupied by a different asset.

Figure 5.10: Set-point control block diagram [18]. The set-point control input, r , is passed in and multiplied by matrices F and N to obtain the new equilibrium condition points. The difference between the new input equilibrium point, x_{eq} , and the feedback state, x , is multiplied by K , and combined with the new control equilibrium point to define the control input, u . The output vector z contains all controlled output variables.

5.5 Subsystems

The asset contains subsystems that are evaluated at every time step. Although the number of subsystems depends on the complexity and/or fidelity of the study, only four subsystems are being utilized with the current scenario. These are, the pheromone subsystem that interacts with the digital pheromones algorithm to choose the next direction of travel for the asset. The controller subsystem that houses the state, control, and gains matrices utilized to command the asset and maintain steady-state flight. The motor subsystem that utilizes reference data to calculate the amount of fuel consumed throughout flight. Lastly, the camera subsystem, it searches the current grid space for the scenario target. As the number of defined subsystems approaches the actual number of asset subsystems, and as the fidelity of each model is improved, the simulation representation accuracy increases. The simulation therefore provides more realistic results that parallel the actual operation of the system during flight tests.

5.5.1 Controller

The set-point controller is utilized to maintain the aircraft at steady-state flight conditions. Heading changes are commanded throughout the entire simulation as the asset maneuvers to different grid spaces on the pheromone map. An important simulation consideration is collision avoidance between assets, this is accomplished by changing the altitude of the asset that takes the same grid space as another asset on the pheromone map. Altitude changes are commanded for collision-avoidance purposes only. The pheromone subsystem checks the adjacent neighboring positions, and verifies that there are none available due to other assets occupying those neighboring positions. The asset chooses a random direction within the defined constraints, and it is commanded to increase altitude by *50 m* for the duration of the maneuver. For this occurrence, two assets occupy the same position on the pheromone map but the difference can be observed in their altitude. If the succeeding set of neighboring spaces are also unavailable due to other assets occupying them, then once again, a random direction is chosen for the maneuver, and the *50 m* altitude change above the defined cruising altitude is maintained. Otherwise, the asset returns to the original altitude for the following maneuver.

The control related matrices were calculated utilizing MATLAB. The controller was tuned to have a quick response but also be slow enough to produce realistic results. The trajectory curves defined in section 5.4.1 were utilized to test the controller response to commands. The trajectories are composed of various individual curves, the types of curves affect the controller response. These changes are more prevalent when inspecting the controller response error, the average error changes when the curves change. The *45 degree* trajectory curve response data can be found in fig. 5.11, and the *90 degree* trajectory curve response data can be found in fig. 5.12.

Figure 5.11: Controller response to a 45 degree heading curve. (a) predefined curve trajectory, and path traveled by UAV asset (b) trajectory error between UAV path and predefined curve (c) velocity changes throughout maneuver (d) changes in Euler angles throughout maneuver (e) altitude change throughout maneuver.

Figure 5.12: Controller response to a 90 degree heading curve. (a) predefined curve trajectory, and path traveled by UAV asset (b) trajectory error between UAV path and predefined curve (c) velocity changes throughout maneuver (d) changes in Euler angles throughout maneuver (e) altitude change throughout maneuver.

5.5.2 Pheromone

The pheromone subsystem receives all asset relevant parameters for interacting with the pheromone map from the XML input file. These parameters include, the starting location of the asset, and all visit pheromone parameters required to create an instance of the defined pheromone. This subsystem accesses the pheromone map, and checks the neighboring spaces' pheromone concentration and occupancy based on the movement constraints of the asset. The current attitude of the asset is taken into consideration to assure the next maneuver is valid based on possible moves within the asset constraints. Based on the result, the subsystem chooses a location to traverse to, and the direction of the move is equivalent to a defined trajectory.

5.5.3 Motor

The powerplant for the MQ-1 Predator is the Rotax 914F four-cylinder engine. It generates 84.5 kW during peak performance running at $5,800 \text{ rpm}$. An image of the engine is displayed in fig. 5.13. Using the graphs provided by the motor datasheet, an equation can be fitted to describe the trends occurring on each plot. For fig. 5.14(a), the approximated equation for fuel consumption in gal/hr is,

$$y = 8 \times 10^{-7}x^2 - 0.0049x + 8.7569 \quad (5.28)$$

with an R^2 value of 0.9935. The equation for fig. 5.14(b), the engine power performance, approximated in kW is,

$$y = 0.0157x - 12.354 \quad (5.29)$$

with a corresponding R^2 value of 0.9943. The equation for fig. 5.14(c) for the engine generated torque in Nm is,

$$y = -5 \times 10^{-6}x^2 + 0.0483x + 4.1595 \quad (5.30)$$

and the R^2 value is approximated to be 0.9993.

Figure 5.13: Rotax 914F four-cylinder engine [32].

Figure 5.14: Rotax 914F engine performance graphs [32]. (a) engine fuel consumption (b) engine power performance (c) engine torque.

5.5.4 Camera

The imaging hardware utilized by the MQ-1 Predator is the AN/AAS-52 Multi-Spectral Targeting System (MTS) manufactured by Raytheon. An image of the MTS is shown in fig. 5.15 to provide a visual representation of the actual system. Even though there is a datasheet for the imaging system containing some relevant information for accurately modeling the MTS, a convenient generalizing approach has been utilized to facilitate the target search for this scenario. The camera subsystem assumes that it is able to image the entire target area of the current asset position, the camera's horizontal and vertical field of view match the grid-space size in the pheromone map. The asset searches for the scenario target throughout the duration of the simulation flight, when the target is within range, a boolean flag is raised and logged in the software for the duration that target is within the camera range.

Figure 5.15: AN/AAS-52 Multi-Spectral Targeting System manufactured by Raytheon [31].

Chapter 6

SCENARIO SIMULATION

Multiple simulation runs are performed with slight variations to some of the parameters to obtain a wide range in output result values. The initial conditions remain the same throughout the various runs, these include having the same starting location and the same target position every time. It is assumed that all vehicles are identical to provide consistent results based only on the pheromone interactions. The variation of parameters relates strictly to the visitation pheromone, therefore, all assets have the same control gains, state and control matrices, and steady-state conditions. All assets begin at the starting location flying at steady-state with an initial heading of zero, which translates to traversing in the north direction. The total simulation time is defined at *800.0 seconds* with interval time-steps of *0.1 seconds*. The target location is approximately *8.5 km* from the starting position. Every asset travels with an initial axial velocity of *36 m/s*, a zero angle of attack, and an altitude of *1,800 m* with an air density value of *1.027 kg/m³*. The target deck contains a definition of all possible targets for the simulation in an XML file. This file contains two defined targets. (1) The simulation target, for this scenario, is a facility target located at coordinates 34.804°N, and 120.603°W and requests an “IMAGING” task when the asset is within range. (2) The second target is not considered a target as dictated by the scenario, it is defined as such for the Scheduler to allow the dynamic movement of the asset while it searches for the simulation target.

Listing 6.1: Target deck XML file

```
1 <TARGETDECK>
2   <TARGET
3     TargetName = "EnemyBuilding"
4     TargetType = "FacilityTarget"
5     TaskType = "IMAGING"
```

```

6         MaxTimes = "1000"
7         Value = "5">
8         <DynamicState
9             DynamicStateType = "STATIC_ECI"
10            ICs = "[34.804;□-120.603;□0.0;□0.0;□0.0;□0.0]">
11        </DynamicState>
12    </TARGET>
13    <TARGET
14        TargetName = "control"
15        TargetType = "FlyingAlong"
16        TaskType = "FLYALONG"
17        MaxTimes = "1000000"
18        Value = "1">
19        <DynamicState
20            DynamicStateType = "STATIC_ECI"
21            ICs = "[34.804;□-120.603;□0.0;□0.0;□0.0;□0.0]">
22        </DynamicState>
23    </TARGET>
24 </TARGETDECK>

```

The variations to the visit pheromone pertain to three different parameters; (1) the update interval, which refers to the wait time, in seconds, before partial evaporation of the pheromone occurs; (2) the pheromone deposit concentration, which is the value an asset assigns to their currently occupied grid-space to alert all other assets of their location; and (3) the pheromone threshold, which describes the lower limit. When the visit pheromone concentration on that grid-space reaches a value below the threshold, the visit concentration is set to zero. The values utilized for the parameter variations are described in table 6.1.

Table 6.1: Simulation scenario variation parameters. There are three different pheromone threshold values for every deposit, three different deposits for every update value, and three different update values for every swarm size.

Number of assets	Update value	Deposit value	Threshold value
1	2, 10, 18	2, 10, 18	1e-02, 1e-10, 1e-18
2	2, 10, 18	2, 10, 18	1e-02, 1e-10, 1e-18
3	2, 10, 18	2, 10, 18	1e-02, 1e-10, 1e-18
4	2, 10, 18	2, 10, 18	1e-02, 1e-10, 1e-18

6.1 Prepare XML Files

Before executing the HSF simulations, the size of the pheromone map must be defined based on the distance from the starting position to the target location. Both positions are defined in coordinate form utilizing latitude and longitude. The component distances between the two points are divided by the defined grid size and rounded down to the nearest integer. Using the ECI reference image from fig. 5.2, the vectors for the two points are approximated, and the angle between the two is calculated. The coordinates are then converted into Cartesian vectors by utilizing the following formulas:

$$x = R_E \sin(90 - \phi) \cos \lambda \quad (6.1)$$

$$y = R_E \sin(90 - \phi) \sin \lambda \quad (6.2)$$

$$z = R_E \cos(90 - \phi) \quad (6.3)$$

where ϕ is the latitude in degrees, and λ is the longitude in degrees. The components between the two points can now be approximated using vectors; and the angle between the two vectors is obtained by utilizing the definition of a dot product,

$$\Omega = \arccos \left(\frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}| |\mathbf{b}|} \right) \quad (6.4)$$

where Ω is in radians. The arc length between the two vectors is then calculated and divided by the desired grid interval to provide an estimate of the pheromone map size,

$$s = R_E \Omega \quad (6.5)$$

where the arc length, s , is assumed to be equivalent to the distance between the two points when using the flat Earth assumption with the NED frame. To convert the asset position from NED to latitude and longitude coordinates, it is assumed that the latitude and longitude conversion formulas for the target are valid for any

Figure 6.1: Kilometers per degree of longitude based on the current latitude. The distance per degree is the largest at the equator, and the smallest at the poles.

asset position. The equivalent distance of meters to degrees remains the same for all latitude values, however, the longitude equivalent varies with changes in latitudes because the circumference decreases as the latitude approaches *90 degrees*. Since most of the search patterns of the swarm occurs near the target location, it is assumed that the conversion error is minimal. For this scenario, the maximum error value for a particular time-step, approximately *100 m*, occurs during the beginning of the simulation when the asset is the farthest from the target. To run all the simulations with different parameters, a batch file is utilized to run a python script that creates the specified XML files, runs HSF, renames the output data files, and moves them to a different directory. A table with important simulation input parameters, and the batch file can be found in Appendix C.

Chapter 7

SIMULATION RESULTS

The conversion value of meters per degree changes for longitude as the asset's latitude changes. Since the conversion factor being utilized for every coordinate calculation is the same as the target location, error is introduced from the beginning of the simulation due to the starting latitude being slightly different from the target latitude. A navigation error is also introduced from the trajectory path following. The asset completes the trajectory maneuver with a small error as discussed in section 5.5.1, and observed in fig. 5.11 and fig. 5.12. The succeeding defined trajectory starts at the current asset location, and therefore as the simulation progresses, error compounds from the different maneuvers.

All the latitude and longitude data has been shifted by -0.00035 , and 0.0023 *degrees* respectively to align the control case data as close as possible to the expected target position. This provides the reader with easier to understand visual data presentation when discussing the asset's path throughout the simulation. A parameter to keep in mind when evaluating the swarm behavior is the pheromone "update" value of the iteration, which can be represented as seconds. It is the required time before updating/evaporating the repelling pheromone. Every asset requires at least 10 seconds to perform a complete maneuver. A lower update value is equivalent to the repelling pheromone evaporating multiple times before one asset maneuver is complete. The "threshold" value also plays a role on how quickly the pheromone map grid space becomes available to new assets. When the "update" and "threshold" values are low, the repelling pheromone may completely evaporate before the asset has completed the current maneuver because evaporation occurs more often, and the pheromone threshold is much smaller.

7.1 Control Case: 1 Asset

The control case consists of a one vehicle simulation, it is utilized as the base performance data to be compared against for all other simulations with swarms of different sizes. The only modified data is the latitude and longitude as previously discussed, all other HSF generated data remains unchanged. The trends between the different cases is discussed but to limit the amount of data being presented to the reader, only the cases with the lowest and highest values for update, deposit, and threshold are displayed. Tables with the data for the other swarm sizes are found in Appendix D. The data for the simulation with an update value of 2 , a deposit value of 2 , and a threshold value of $1e-02$ is discussed with fig. 7.1 and fig. 7.2. The mean distance the asset covered from the marked target is 363.49 m , with a maximum distance of 803.32 m at the end of the simulation. The total fuel consumption for the 800 sec simulation

Figure 7.1: Control case - lowest parameters, target distance through time. (left) Target distance from start of simulation. (right) Target distance from 260 sec to the end of the simulation with the mean distance marked by a dashed red line, and the max distance is marked by a red circle.

is 0.550 gal. According to General Atomics, the MQ-1 has a flight endurance of up to 40 hours with a 100 gal fuel tank. Assuming that the aircraft travels at steady-state flight conditions for the duration of an 800 sec mission, the total fuel consumed is 0.555 gal. The fuel consumption graph, fig. 7.3, displays reoccurring behavior for the

Figure 7.2: Control case - lowest parameters, asset navigation through pheromone map. (left) asset path traveled from start position to target location from start to end of simulation. (right) asset path traveled at vicinity of target location to the end of the simulation.

Figure 7.3: Control case - lowest parameters, asset fuel consumption and time-on-target.

majority of the simulation which coincides with the same turning maneuvers being performed. Since the pheromone value parameters utilized for the simulation are the smallest, the previously visited pheromone map grid-spaces become available once more by the time the asset returns to image the target again. Once at the target location, it is a possible that the asset may randomly travel in any direction due to all neighboring grids having the same attractive pheromone concentration. In fig. 7.2, the asset maintained a left turn trajectory until the target location was once again reached, then the asset chose to maneuver to a different grid-space. Using the lowest parameters for the repelling pheromone allows for the asset to remain near the target,

Figure 7.4: Control case - lowest parameters, asset dynamic state. (a) body frame axial and lateral velocity (b) body frame normal velocity, and asset altitude (c) Euler angles (d) angle rates.

this is ideal for providing constant updates on a fire.

The data for the control case with the highest parameter values have an update of 18 , a deposit value of 18 , and a threshold set at $1e-18$. The distance data is displayed in fig. 7.5, and the asset path traveled is in fig. 7.6. The mean distance between the asset and the target is 823.01 m which is significantly larger than the mean distance from the data with the smallest parameters. The maximum distance is $1,749.71\text{ m}$ and also occurs toward the end of the simulation. The asset moves farther away from the target as the visitation pheromone nearby is not completely evaporated. Once the pheromone evaporates, the asset travels back to the target location to search the site. In fig. 7.6, the asset is observed traveling to the target and subsequently

Figure 7.5: Control case - highest parameters, target distance through time. (left) Target distance from start of simulation. (right) Target distance from 260 sec to the end of the simulation with the mean distance marked by a dashed red line, and the max distance is marked by a red circle.

surveying the areas close by. Since the visit pheromone requires a longer period to disappear, the asset is allowed to search the surrounding grid-spaces. This is useful for scenarios where the target position only defines a possible location, and the asset

is required to search the surrounding areas for the target. In the case of the lost hiker, the target coordinates can be the last known sighting or the possible hiker location. Similar to the data discussed earlier, the fuel consumption for the entire simulation is *0.532 gal*. Even though the asset performs maneuvers to reach different locations than in previous simulations, the set-point controller limits the dynamic state changes to maintain conditions at steady-state flight. The dynamic state data for the asset

Figure 7.6: Control case - highest parameters, asset navigation through pheromone map. (left) Asset path traveled from start position to target location from start to end of simulation. (right) Asset path traveled at vicinity of target location to the end of the simulation.

is presented in fig. 7.8. The path traveled by the asset is much different from the earlier simulation data presented because the pheromone requires a longer period of time to fully evaporate, and therefore the dynamic state of the asset through time is also much different. A similar trend observed between the control case data and all other output data is the angular rates, specially the roll rate, they are unusually high because the controller is too responsive and needs to be tuned. The $\mathbf{Q}$ and $\mathbf{R}$ matrices need to be adjusted to calculate an optimal set of gains. For the modeling

Figure 7.7: Control case - highest parameters, asset fuel consumption and time-on-target.

Figure 7.8: Control case - highest parameters, asset dynamic state. (a) Body frame axial and lateral velocity (b) Body frame normal velocity, and asset altitude (c) Euler angles (d) angle rates.

and simulation of the swarm, the current controller is acceptable.

A summary of all the data outputs from the different control cases simulated can be found in table 7.1. The first column is the update value for the simulation, which corresponds to the amount of time that is required to elapse before an evaporation event occurs. The second column represents the deposit value, also known as the pheromone concentration. The third column displays the parameter description obtained from the simulated data. The last three columns define the threshold value, which is the limit a pheromone concentration may achieve before total evaporation.

Table 7.1: Control case output data.

Update	Deposit	Parameter	Thrsh 1e-02	Thrsh 1e-10	Thrsh 1e-18
2	2	Max distance	803.32 m	585.71 m	805.22 m
2	2	Mean distance	363.49 m	353.61 m	395.68 m
2	2	Fuel consumed	0.5496 gal	0.5516 gal	0.5439 gal
2	10	Max distance	701.01 m	585.71 m	953.69 m
2	10	Mean distance	401.61 m	353.61 m	418.12 m
2	10	Fuel consumed	0.5478 gal	0.5516 gal	0.5406 gal
2	18	Max distance	701.01 m	786.07 m	978.58 m
2	18	Mean distance	415.75 m	360.96 m	455.56 m
2	18	Fuel consumed	0.5404 gal	0.5633 gal	0.5420 gal
10	2	Max distance	803.32 m	1342.21 m	1515.37 m
10	2	Mean distance	363.49 m	660.76 m	673.46 m
10	2	Fuel consumed	0.5496 gal	0.5331 gal	0.5353 gal
10	10	Max distance	775.30 m	1515.37 m	1532.38 m
10	10	Mean distance	390.33 m	673.46 m	699.09 m
10	10	Fuel consumed	0.5463 gal	0.5353 gal	0.5355 gal
10	18	Max distance	1293.83 m	1515.37 m	1532.38 m

10	18	Mean distance	591.77 m	673.46 m	699.09 m
10	18	Fuel consumed	0.5416 gal	0.5353 gal	0.5355 gal
18	2	Max distance	954.32 m	1468.85 m	1495.90 m
18	2	Mean distance	449.65 m	742.16 m	784.80 m
18	2	Fuel consumed	0.5394 gal	0.5329 gal	0.5322 gal
18	10	Max distance	941.11 m	1491.96 m	1825.56 m
18	10	Mean distance	473.72 m	754.01 m	923.01 m
18	10	Fuel consumed	0.5383 gal	0.5310 gal	0.5318 gal
18	18	Max distance	1627.46 m	1491.96 m	1749.71 m
18	18	Mean distance	795.80 m	751.75 m	823.01 m
18	18	Fuel consumed	0.5421 gal	0.5311 gal	0.5320 gal

7.2 Case: 2 Assets

The second case consists of a two-vehicle swarm. Both assets are launched from the same location at the same time. The assets use the pheromone gradient to climb in the direction of the target. As the size of the swarm increases to two-vehicles, the mean distance for the lowest parameter case increases to *576.03 m*, and the maximum distance achieved for the overall swarm is *1,310.21 m*. The average distance change can be observed through fig. 7.10, it shows how the search area is distributed among each asset. Similar to the control case with the lowest parameters, each asset wants to fly in a recurring pattern in the vicinity of the target. Since there is more than one vehicle trying to image the same target location, each asset maneuvers to its own general area to survey. Since the assets try to remain in a small area near the target, the Euler angles show a brief unnatural change. The controller response

Figure 7.9: Case: 2 assets - lowest parameters, target distance through time. Target distance from 260 sec to the end of the simulation with the mean distance marked by a dashed red line, and the max distance is marked by a red circle.

Figure 7.10: Case: 2 assets - lowest parameters, asset navigation through pheromone map. (left) Asset path traveled from start position to target location from start to end of simulation. (right) Asset path traveled at vicinity of target location to the end of the simulation.

was discussed in the control case section, the roll angles of fig. 7.11 in particular show a big spike that manifests only with the low parameter cases. The spike represents an omission for the asset when selecting a turn based on the asset's current attitude. The bigger spike observed in the asset one Euler angles is due to a near *90 degree* angle trajectory traveled in a very short period of time. The smaller spike observed in the asset two Euler angles is due to consecutive *90 degree* maneuvers in the same direction. Even though the dynamic states may be unrealistic, the simulation results from HSF are still within the given constraints and therefore those schedules are not rejected.

Figure 7.11: Case: 2 assets - lowest parameters, asset dynamic state. (a) Asset 1 Euler angles (b) Asset 2 Euler angles.

The data with higher parameters shows a greater variation when fig. 7.14, the path traveled by the assets is observed. The overall area searched by the assets is greater because the repel pheromone requires a longer period of time to evaporate. This allows the assets to explore the target's surroundings without having to return to the visited grid-spaces too often. The number of assets also affects the size of the searched area. The same pheromone map is utilized by all assets; therefore, when

Figure 7.12: Case: 2 assets - lowest parameters, asset time-on-target.

more vehicles are introduced into the swarm, the number of available grid-spaces close to the target will soon be visited, forcing the assets to expand the search farther out. As the number of vehicles in the swarm increases, the search area generally increases as well. For this particular case, the mean distance is $1,006.08\text{ m}$ with a maximum distance of 2072.17 m . The update and deposit value provide a similar trend change

Figure 7.13: Case: 2 assets - highest parameters, target distance through time. Target distance from 260 sec to the end of the simulation with the mean distance marked by a dashed red line, and the max distance is marked by a red circle.

Figure 7.14: Case 2 assets- highest parameters, asset navigation through pheromone map. (left) Asset path traveled from start position to target location from start to end of simulation. (right) Asset path traveled at vicinity of target location to the end of the simulation.

Figure 7.15: Case: 2 assets - highest parameters, asset dynamic state. (a) Asset 1 Euler angles (b) Asset 2 Euler angles.

to the swarm behavior. The most effective adjustments for an area search resides in the combination of small threshold limits, and a higher deposit or update value to allow the repel pheromone to not evaporate quickly. If the mission requires constant

imaging of a target at a known location, then lower parameters are desired. When the repelling pheromone evaporates quickly, the assets are allowed to return to the target and image the location more often. A table with all the summarized simulation values for the two asset swarm can be found in table 7.2 at the end of this section.

Figure 7.16: Case: 2 assets - highest parameters, asset time-on-target.

Table 7.2: 2 asset swarm distance summary data.

Update	Deposit	Parameter	Thrsh 1e-02	Thrsh 1e-10	Thrsh 1e-18
2	2	Max distance	1310.21 m	945.92 m	1141.05 m
2	2	Mean distance	576.03 m	491.27 m	536.49 m
2	10	Max distance	1042.89 m	942.54 m	942.54 m
2	10	Mean distance	424.43 m	470.50 m	492.95 m
2	18	Max distance	1062.56 m	942.54 m	942.54 m
2	18	Mean distance	409.49 m	470.50 m	492.95 m
10	2	Max distance	1013.17 m	1737.64 m	1990.00 m
10	2	Mean distance	477.42 m	882.61 m	902.04 m
10	10	Max distance	1167.99 m	2219.30 m	2338.17 m
10	10	Mean distance	535.14 m	1114.19 m	1225.35 m
10	18	Max distance	1072.02 m	1937.57 m	1955.30 m

10	18	Mean distance	565.77 m	895.91 m	949.03 m
18	2	Max distance	1243.47 m	1926.65 m	2445.34 m
18	2	Mean distance	605.17 m	985.36 m	1335.85 m
18	10	Max distance	1510.98 m	2165.83 m	2110.95 m
18	10	Mean distance	700.59 m	1120.84 m	1286.03 m
18	18	Max distance	1654.25 m	2007.87 m	1956.30 m
18	18	Mean distance	757.19 m	1198.38 m	1017.41 m

7.3 Case: 3 Assets

The three-vehicle swarm follows a similar pattern pattern as the previous cases except that the assets all fall under a similar trajectory where they all navigate near the target location. As a result, the average distance for the three-asset swarm is smaller than the two-asset swarm, where each vehicle separates after reaching the target and searches it's respective area. The mean distance is 423.89 m , and the max distance achieved is 958.55 m . As it can be observed in the time-on-target graphs, fig. 7.17, the three-asset swarm spends the majority of the flight time imaging the target.

Figure 7.17: Case: 3 assets - lowest parameters, asset time-on-target.

Figure 7.18: Case: 3 assets - lowest parameters, target distance through time. Target distance from 260 sec to the end of the simulation with the mean distance marked by a dashed red line, and the max distance is marked by a red circle.

Figure 7.19: Case: 3 assets - lowest parameters, asset navigation through pheromone map. (left) Asset path traveled from start position to target location from start to end of simulation. (right) Asset path traveled at vicinity of target location to the end of the simulation.

(a)

(b)

(c)

Figure 7.20: Case: 3 assets - lowest parameters, asset dynamic state. (a) Asset 1 Euler angles (b) Asset 2 Euler angles (c) Asset 3 Euler angles.

type of behavior is important for targets at a known location that require constant surveillance. Similar to the 2 asset case, the low parameters simulation exhibit some unusual spikes due to omissions. For search-type missions, the higher parameter cases are preferred, where the target location merely serves as an area marker that the assets can use to search around. The flight trajectories for the assets are displayed in fig. 7.22, the assets avoid the visited grid-spaces. The average distance from the target is $1,647.76\text{ m}$, and the maximum is $2,587.89\text{ m}$.

Figure 7.21: Case: 3 assets - highest parameters, target distance through time. (left) Target distance from start of simulation. (right) Target distance from 260 sec to the end of the simulation with the mean distance marked by a dashed red line, and the max distance is marked by a red circle.

Figure 7.22: Case 3 assets- highest parameters, asset navigation through pheromone map. (left) Asset path traveled from start position to target location from start to end of simulation. (right) Asset path traveled at vicinity of target location to the end of the simulation.

Figure 7.23: Case: 3 assets - highest parameters, asset time-on-target.

(a)

(b)

(c)

Figure 7.24: Case: 3 assets - highest parameters, asset dynamic state. (a) Asset 1 Euler angles (b) Asset 2 Euler angles (c) Asset 3 Euler angles.

Table 7.3: 3 asset swarm distance summary data.

Update	Deposit	Parameter	Thrsh 1e-02	Thrsh 1e-10	Thrsh 1e-18
2	2	Max distance	958.55 m	1176.60 m	1333.80 m
2	2	Mean distance	423.89 m	656.14 m	663.03 m
2	10	Max distance	1122.93 m	1080.78 m	1300.82 m
2	10	Mean distance	503.57 m	606.36 m	622.36 m
2	18	Max distance	1008.77 m	1369.32 m	1080.78 m
2	18	Mean distance	436.26 m	689.61 m	648.50 m
10	2	Max distance	1486.83 m	1903.48 m	1681.95 m
10	2	Mean distance	704.79 m	1087.25 m	1253.78 m
10	10	Max distance	1327.79 m	1960.43 m	2373.73 m
10	10	Mean distance	677.98 m	1074.89 m	1371.82 m
10	18	Max distance	1618.33 m	1954.59 m	1936.83 m
10	18	Mean distance	706.68 m	1124.98 m	1313.21 m
18	2	Max distance	1778.72 m	2399.15 m	2587.89 m
18	2	Mean distance	816.16 m	1445.67 m	1647.76 m
18	10	Max distance	1662.46 m	2399.15 m	2587.89 m
18	10	Mean distance	906.70 m	1445.67 m	1647.76 m
18	18	Max distance	1832.96 m	2399.15 m	2587.89 m
18	18	Mean distance	979.13 m	1445.67 m	1647.76 m

7.4 Case: 4 Assets

The four-vehicle swarm is the last case simulated for the study. Similar to the smaller swarm sizes, the lowest parameter case stays within the vicinity of the target location

because the repelling pheromone evaporates quickly. The trajectory path of asset 4, purple path, in fig. 7.27 provides an idea of the evaporation rate. Asset 4 maneuvers to take the trajectory following asset 1 because the majority, if not all, of the repelling pheromone has already evaporated and the attractive pheromone gradient at that grid-space is the highest. Since the assets maintain a recurring flight trajectory around the target, the swarm provides a semi constant coverage of the target location. The recurring maneuvers allow the assets to image the target as often as possible. Even though the same maneuvers are constantly executed, the roll angle constraint is not violated, with the exception of the controller response spikes in the low parameter cases due to omissions. The average swarm distance achieved is 451.44 m , and a maximum distance of $1,000.61\text{ m}$. The higher parameter case provides the widest area search and coverage around the target with a mean distance of $1,930.17\text{ m}$, and a max distance of $3,499.66\text{ m}$. The swarm is able to survey almost the entire surrounding area as seen in fig. 7.31 due to the repelling pheromone evaporating slower.

Figure 7.25: Case: 4 assets - lowest parameters, asset time-on-target.

Figure 7.26: Case: 4 assets - lowest parameters, target distance through time. Target distance from 260 sec to the end of the simulation with the mean distance marked by a dashed red line, and the max distance is marked by a red circle.

Figure 7.27: Case: 4 assets - lowest parameters, asset navigation through pheromone map. (left) Asset path traveled from start position to target location from start to end of simulation. (right) Asset path traveled at vicinity of target location to the end of the simulation.

Figure 7.28: Case: 4 assets - lowest parameters, asset dynamic state. (a) Asset 1 Euler angles (b) Asset 2 Euler angles (c) Asset 3 Euler angles (d) Asset 4 Euler angles.

Figure 7.29: Case: 4 assets - highest parameters, asset time-on-target.

Figure 7.30: Case: 4 assets - highest parameters, target distance through time. (left) Target distance from start of simulation. (right) Target distance from 260 sec to the end of the simulation with the mean distance marked by a dashed red line, and the max distance is marked by a red circle.

Figure 7.31: Case 4 assets- highest parameters, asset navigation through pheromone map. (left) Asset path traveled from start position to target location from start to end of simulation. (right) Asset path traveled at vicinity of target location to the end of the simulation.

Figure 7.32: Case: 4 assets - highest parameters, asset dynamic state. (a) Asset 1 Euler angles (b) Asset 2 Euler angles (c) Asset 3 Euler angles (d) Asset 4 Euler angles.

Table 7.4: 4 asset swarm distance summary data.

Update	Deposit	Parameter	Thrsh 1e-02	Thrsh 1e-10	Thrsh 1e-18
2	2	Max distance	1037.08 m	1393.03 m	1324.13 m
2	2	Mean distance	453.35 m	653.28 m	690.83 m
2	2	Fuel consumed	0.5546 gal	0.5344 gal	0.5360 gal
2	10	Max distance	1290.78 m	1475.72 m	1378.04 m

2	10	Mean distance	563.56 m	626.55 m	735.39 m
2	10	Fuel consumed	0.5558 gal	0.5373 gal	0.5351 gal
2	18	Max distance	1278.34 m	1494.22 m	1494.22 m
2	18	Mean distance	495.99 m	749.03 m	793.98 m
2	18	Fuel consumed	0.5421 gal	0.5348 gal	0.5383 gal
10	2	Max distance	1642.15 m	2192.67 m	1797.94 m
10	2	Mean distance	740.54 m	1179.95 m	1526.83 m
10	2	Fuel consumed	0.5367 gal	0.5335 gal	0.5314 gal
10	10	Max distance	1642.15 m	2129.93 m	1797.94 m
10	10	Mean distance	743.74 m	1220.69 m	1274.82 m
10	10	Fuel consumed	0.5367 gal	0.5293 gal	0.5344 gal
10	18	Max distance	1822.66 m	2129.93 m	2338.07 m
10	18	Mean distance	836.42 m	1222.24 m	1293.86 m
10	18	Fuel consumed	0.5354 gal	0.5312 gal	0.5331 gal
18	2	Max distance	1944.15 m	2137.75 m	2137.75 m
18	2	Mean distance	861.25 m	1561.43 m	1619.05 m
18	2	Fuel consumed	0.5349 gal	0.5331 gal	0.5334 gal
18	10	Max distance	1842.30 m	3426.12 m	2965.99 m
18	10	Mean distance	966.72 m	1941.32 m	1634.67 m
18	10	Fuel consumed	0.5315 gal	0.5319 gal	0.5312 gal
18	18	Max distance	1986.78 m	3499.66 m	2137.75 m
18	18	Mean distance	1041.03 m	1927.35 m	1753.97 m
18	18	Fuel consumed	0.5303 gal	0.5312 gal	0.5345 gal

CONCLUSION AND FUTURE WORK

8.1 Conclusion

Utilizing digital pheromones to control swarm behavior allows for the search of the area surrounding a defined target location such as searching for a missing hiker at the last known location; or the imaging of the defined location as often as possible such as constant image updates of a fire. Varying the repel pheromone parameters from low to higher values translates to increased revisit time of the target location, and surrounding area surveying respectively. One of the three parameters varied consists of the update value, which defines the number of seconds before all pheromones on the pheromone map are propagated. The parameter value changes are 2, 10, and 18. The second parameter consists of the deposit value, referring to the amount of pheromone concentration introduced into the pheromone map by the asset. The deposit values utilized consists of 2, 10, and 18. The last parameter varied is the threshold value which provides a lower limit for the repel pheromone. When the deposit value evaporates to less than the threshold value, the repel concentration for that grid-space becomes zero. The threshold limits utilized are $1e-02$, $1e-10$, and $1e-18$.

As the swarm grows in size, the total time-on-target, or the surrounding area surveyed, also increases depending on the pheromone. The four-vehicle swarm provides an almost constant imaging of the target when utilizing the lowest parameters. Disadvantages occur in the constant circular flight trajectory because all assets fly a similar path for the majority of the time. The four-vehicle swarm with the highest parameters provides the widest area surveyed by the assets. The repel pheromone

requires longer to evaporate and therefore the assets visit each surrounding grid-space once before expanding the search outward to neighboring spaces.

8.2 Future Work

There are various improvements that can be incorporated into the simulation models to increase the fidelity of the HSF outputs, ranging from new environmental and subsystem models, to expanding on the current models being utilized. A new model that can be included is the wind model as part of the environment node. An accurate wind model for the target location can increase the fidelity of the results by better approximating real physical environmental conditions and turbulence that the asset may encounter. Along with the wind model, a new reference frame would need to be included. The wind reference frame is similarly defined to the body frame but the x-axis points in the direction of the wind.

The generalization that utilized a constant angle of attack of 0 degrees for the state and control matrices, $\mathbf{A}$ and $\mathbf{B}$, can be improved. A lookup table with the different stability and control derivatives for various angles of attack and sideslip angles can be included. The stability and control derivatives for angles not listed on the table can be interpolated. Along with the lookup table, a function for generating optimal trajectories between two locations for a 3D pheromone map can be included. The Frenet-Serret formulae can then be implemented, and utilized to its full potential with 3D maneuvers.

Some new subsystem models to be included into the asset to yield better results include the IMU model that approximates the states of the asset based on the data obtained from the accelerometer and gyroscope. Utilizing the IMU to obtain dynamic state data is a more realistic approach than calculating and feeding the full state into the feedback control loop. Like any mechanical sensor, the IMU is also affected

by random noise which generates a noisy reading. Noise affecting the state signals can then be minimized by introducing a Kalman filter subsystem into the asset. An adjustment to the controller can also be made to improve the fidelity of the overall results. These adjustments provide optimal asset performance based on the mechanical limitation of the actual physical system. Another improvement to the simulation comes as the addition of image processing to the camera subsystem. Each grid-space on the pheromone map can be assigned images corresponding to their area. The camera subsystem can then process each image until the target is identified within the image assigned to the target grid-space.

An image recognition package that could be incorporated into the camera subsystem in the future is the AI software, Project Maven, that Google has been working on for the Department of Defense. The software can autonomously categorize objects in images without human interaction. This is ideal for UAV reconnaissance missions that can generate massive amounts of data composed of images and motion video. This scenario can also be expanded and utilized in the future to test swarming algorithm interactions or to test vehicle component models as part of OFFensive Swarm Enabled Tactics (OFFSET). This is a Defense Advanced Research Projects Agency (DARPA) RFP/RFI, that was initially started in February 2017. A new amended request began in March 2018 is currently active as of the writing of this paper.

BIBLIOGRAPHY

- [1] Creech Air Force Base. sun setting the mq-1 predator: The final salute.
<http://www.creech.af.mil/News/Article-Display/Article/1462948/sun-setting-the-mq-1-predator-the-final-salute/>. Accessed:
2018-03-29.
- [2] MathWorks. <https://www.mathworks.com>. Accessed: 2017-12-14.
- [3] Public Domain Aeronautical Software. properties of the u.s. standard atmosphere 1976. <http://www.pdas.com/atmos.html>. Accessed: 2017-12-06.
- [4] U.S. Air Force. mq-1b predator. <http://www.af.mil/About-Us/Fact-Sheets/Display/Article/104469/mq-1b-predator/>. Accessed:
2017-09-26.
- [5] U.S. Standard Atmosphere 1976. United States Committee on Extension to the Standard Atmosphere (COESA).
https://ccmc.gsfc.nasa.gov/modelweb/atmos/us_standard.html. Accessed:
2017-12-06.
- [6] AFCESA. Airfield Planning and Design Criteria for Unmanned Aircraft Systems (UAS). ETL 09-1, USAF, sep 2009.
- [7] M. L. Anderson, J. Rios, D. Stone, J. Cuany, C. Rasmussen, and L. Hale. Networked cooperative swarm system for area denial operations. In *AIAA Infotech@ Aerospace*, page 1494. 2016.
- [8] B. Butler. Dynamic model creation and scripting support in the horizon simulation framework. Master’s thesis, California Polytechnic State University, San Luis Obispo, 2012.

- [9] G. Cai, B. M. Chen, and T. H. Lee. *Unmanned rotorcraft systems*. Springer Science & Business Media, 2011.
- [10] H. Chen, J. Sun, K. Li, and M. Wang. Autonomous spacecraft swarm formation planning using artificial field based on nonlinear bifurcation dynamics. In *AIAA Guidance, Navigation, and Control Conference*, page 1269, 2017.
- [11] T. Chi, H. Cheng, J. Page, and N. Ahmed. Evolving swarm of uavs. *Advances in aircraft and spacecraft science*, 1(2):219–232, 2014.
- [12] M. V. Cook. *Flight dynamics principles: a linear systems approach to aircraft stability and control*. Butterworth-Heinemann, 2012.
- [13] Defense Airborne Reconnaissance Office. Uav annual report fy 1996. Technical report, Department of Defense, nov 1996.
- [14] Department of the Air Force. Designating and naming military aerospace vehicles. Technical report, Department of Defense, may 2014.
- [15] M. Drela and H. Youngren. Athena vortex lattice. *Software Package, Ver, 3.36*, 2017.
- [16] R. Falconi and C. Melchiorri. A decentralized control algorithm for swarm behavior and obstacle avoidance in unknown environments. *IFAC Proceedings Volumes*, 41(1):44–49, 2008.
- [17] R. Finck, A. F. F. D. L. (US), and D. Hoak. *USAF stability and control DATCOM*. Engineering Documents, 1978.
- [18] J. P. Hespanha. Lecture notes on lqr/lqg controller design. *Knowledge Creation Diffusion Utilization*, 2005.

- [19] X. Hu, R. C. Eberhart, and Y. Shi. Swarm intelligence for permutation optimization: a case study of n-queens problem. In *Swarm intelligence symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*, pages 243–246. IEEE, 2003.
- [20] S. Iovino, A. R. Vetrella, G. Fasano, D. Accardo, and A. Savvaris. Implementation of a distributed flocking algorithm with obstacle avoidance capability for uav swarming. In *AIAA Information Systems-AIAA Infotech@Aerospace*, page 0878. 2017.
- [21] V. K. Kalivarapu. *Improving solution characteristics of particle swarm optimization through the use of digital pheromones, parallelization, and graphical processing units (GPUs)*. Iowa State University, 2008.
- [22] Z. M. Li. Modeling and simulation of autonomous thermal soaring with horizon simulation framework. Master’s thesis, California Polytechnic State University, San Luis Obispo, 2010.
- [23] S. MacLean. Modeling and simulation of a sounding rocket active stabilization system. Master’s thesis, California Polytechnic State University, San Luis Obispo, 2017.
- [24] A. Maria. Introduction to modeling and simulation. In *Proceedings of the 29th Conference on Winter Simulation, WSC '97*, pages 7–13, Washington, DC, USA, 1997. IEEE Computer Society.
- [25] D. Morgan, S.-J. Chung, and F. Y. Hadaegh. Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and model predictive control. In *AIAA guidance, navigation, and control conference*, page 0599, 2015.

- [26] Y. Nagao and K. Uchiyama. Formation flight of fixed-wing uavs using artificial potential field. In *29th Congress of the International Council of the Aerospace Sciences*, 2014.
- [27] E. V. Ntagiou, P. Palmer, C. Iacopino, N. Policella, and A. Donati. Coverage planning for agile eo constellations using ant colony optimisation. In *14th International Conference on Space Operations*, page 2448, 2016.
- [28] C. O’Connor. Horizon: A system modeling and simulation framework for systems engineering utility analysis. Master’s thesis, California Polytechnic State University, San Luis Obispo, 2007.
- [29] N. M. Patrikalakis and T. Maekawa. *Shape interrogation for computer aided design and manufacturing*. Springer Science & Business Media, 2009.
- [30] S. Raczynski. *Modeling and Simulation: The Computer Science of Illusion*. RSP. Wiley, Somerset, 2006.
- [31] Raytheon Space and Airborne Systems. *AN/AAS-52 Multi-Spectral Targeting System*, 3 2005.
- [32] Rotax Aircraft Engines. *Engine Type 914*, 2017.
- [33] J. A. Sauter, R. Matthews, H. Van Dyke Parunak, and S. A. Brueckner. Performance of digital pheromones for swarming vehicle control. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 903–910. ACM, 2005.
- [34] J. A. Sauter, R. S. Matthews, H. V. D. Parunak, and S. A. Brueckner. Effectiveness of digital pheromones controlling swarming vehicles in military scenarios. *JACIC*, 4(5):753–769, 2007.

- [35] D. A. Spencer. Automated trajectory control using artificial potential functions to target relative orbits. *Journal of Guidance, Control, and Dynamics*, pages 2142–2148, 2016.
- [36] USAF. Standard atmosphere, 1976. *NASA TMX-74335*, pages 12–15, 1976.
- [37] J. G. Warner and S. D. Vtipil. Earth observing satellite orbit design via particle swarm optimization. In *AIAA/AAS Astrodynamics Specialist Conference*, page 4428, 2014.
- [38] M. Yost. An iteration on the horizon simulation framework to include .net and python scripting. Master’s thesis, California Polytechnic State University, San Luis Obispo, 2016.
- [39] R. Zou, V. K. Kalivarapu, S. Bhattacharya, E. H. Winer, and J. Oliver. Standard particle swarm optimization on source seeking using mobile robots. In *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 0897, 2015.

APPENDICES

Appendix A

ENVIRONMENTAL MODEL DATA

A.1 HSF Atmosphere Model

Table A.1: 1976 U.S. Standard Atmosphere data generated by HSF for various geometric altitudes.

Altitude (m)	Temperature (K)	Pressure (Pa)	Density (kg/m ³)
0	2.88150e+02	1.01325e+05	1.22500e+00
4,000	2.62166e+02	6.16604e+04	8.19346e-01
8,000	2.36215e+02	3.56515e+04	5.25785e-01
12,000	2.16650e+02	1.93993e+04	3.11937e-01
16,000	2.16650e+02	1.03527e+04	1.66469e-01
20,000	2.16650e+02	5.52921e+03	8.89084e-02
24,000	2.20560e+02	2.97167e+03	4.69366e-02
28,000	2.24527e+02	1.61614e+03	2.50754e-02
32,000	2.28490e+02	8.89024e+02	1.35545e-02
36,000	2.39283e+02	4.98495e+02	7.25749e-03
40,000	2.50351e+02	2.87125e+02	3.99540e-03
44,000	2.61405e+02	1.69484e+02	2.25867e-03
48,000	2.70650e+02	1.02286e+02	1.31658e-03
52,000	2.69029e+02	6.22084e+01	8.05541e-04
56,000	2.58017e+02	3.73576e+01	5.04394e-04
60,000	2.47018e+02	2.19554e+01	3.09636e-04

A.2 PDAS Atmosphere Model

Table A.2: 1976 U.S. Standard Atmosphere data generated by PDAS for various geometric altitudes.

Altitude (m)	Temperature (K)	Pressure (Pa)	Density (kg/m ³)
0	288.1	1.013E+5	1.225E+0
4,000	262.2	6.166E+4	8.193E-1
8,000	236.2	3.565E+4	5.258E-1
12,000	216.6	1.940E+4	3.119E-1
16,000	216.6	1.035E+4	1.665E-1
20,000	216.6	5.529E+3	8.891E-2
24,000	220.6	2.972E+3	4.694E-2
28,000	224.5	1.616E+3	2.508E-2
32,000	228.5	8.890E+2	1.355E-2
36,000	239.3	4.985E+2	7.257E-3
40,000	250.4	2.871E+2	3.995E-3
44,000	261.4	1.695E+2	2.259E-3
48,000	270.6	1.023E+2	1.317E-3
52,000	269.0	6.221E+1	8.055E-4
56,000	258.0	3.736E+1	5.044E-4
60,000	247.0	2.196E+1	3.096E-4
64,000	236.0	1.260E+1	1.860E-4
68,000	225.1	7.051E+0	1.091E-4
72,000	214.3	3.835E+0	6.236E-5
76,000	206.4	2.033E+0	3.430E-5
80,000	198.6	1.052E+0	1.845E-5

Appendix B

UAV MODEL DATA

Table B.1: Lift coefficient, C_L , data obtained from the Digital DATCOM and AVL software.

α (deg)	DATCOM	AVL
-2	-0.0157	-0.03004
-1	0.0771	0.07905
-0.5	0.1242	0.13359
0	0.1718	0.18812
0.5	0.2198	0.24265
1	0.2681	0.29715
2	0.3660	0.40609
4	0.5660	0.62355
8	0.9849	1.05560
12	1.3636	1.48169
14	1.5054	1.69171
16	1.6124	1.89930
18	1.6060	2.10416
20	1.1240	2.30600

Table B.2: Max roll angle, ϕ_{max} data calculated using the corresponding reference airspeed and a stall speed of *27.2 m/s*.

Airspeed (m/s)	Max Roll, ϕ_{max} (deg)
27.2	0
28.0	19.3
29.0	28.4
30.0	34.7
31.0	39.7
32.0	43.7
33.0	47.2
34.0	50.2
35.0	52.8
36.0	55.2
37.0	57.3
38.0	59.2
39.0	60.9
40.0	62.5

Table B.3: MQ-1 Predator model data.

Description	Data
Mach	0.108 (local)
Airspeed	36 m/s
Gravity, g	9.80665 m/s^2
Mass, m	800 kg
I_{xx}	6,800 $kg \cdot m^2$
I_{yy}	11,500 $kg \cdot m^2$
I_{zz}	18,500 $kg \cdot m^2$
I_{xz}	0 $kg \cdot m^2$
Airfoil	NACA 2414
Wingspan, b	16.84 m
Root chord	1.10 m
Tip chord	0.40 m
Wing Reference Area, S_{ref}	12.84 m^2
Aileron length	3.465 m
Propeller radius	0.9144 m
Propeller efficiency	0.85

B.1 Digital DATCOM Data

Listing B.1: Datcom Input File

```
1 CASEID MQ-1 Predator: Body, Wings
2 DIM M
3 DERIV RAD
4 DAMP
5 PART
6 *****
7 * Flight Conditions *
8 *****
9 $FLTCON WT=800.0,
10          NMACH=1.0, MACH(1)=0.108,
11          NALT=1.0, ALT(1)=1800.0, NALPHA=9.0,
12          ALSCHD(1)=-2.0, -1.0, -0.5, 0.0, 0.5, 1.0, 2.0, 4.0, 8.0$
13 *****
14 * Reference Parameters *
15 *****
16 $OPTINS SREF=12.84, CBARR=1.10, BLREF=16.84$
17 *****
18 * Group II          Synthesis Parameters *
19 *****
20
21 $SYNTHS XCG=4.044, ZCG=0.0,
22          XW=3.769, ZW=-0.225, ALIW=0.0,
23          XV=6.891, YV=0.305, ZV=-0.200,
24          VERTUP=.FALSE., PHIV =55.0,
25          XVF=6.891, ZVF=-0.200,
26          VERTUP=.FALSE., HINAX=7.069$
27 *****
28 * Body Configuration Parameters *
29 *****
30 $BODY NX=12.0,
31        X(1) =0.000, 0.000, 0.384, 0.682, 0.938, 1.194,
32              1.492, 3.070, 3.497, 7.419, 8.144, 8.247,
33        R(1) =0.000, 0.061, 0.392, 0.508, 0.533, 0.549,
34              0.549, 0.381, 0.371, 0.371, 0.102, 0.000,
35        ZU(1)=-0.270, -0.260, 0.063, 0.270, 0.396, 0.468,
36              0.468, 0.288, 0.288, 0.288, 0.135, 0.000,
37        ZL(1)=-0.270, -0.280, -0.585, -0.585, -0.585, -0.585,
38              -0.585, -0.342, -0.315, -0.315, -0.135, 0.000,
39        ITYPE=1.0$
40 *****
41 *      Wing Planform Variables *
42 *****
43 $WGPLNF CHRDR=1.13, CHRDT=0.40,
44          SSPN=8.42, SSPNE=8.039,
45          SAVSI=2.49, SAVSO=-2.49, CHSTAT=0.0,
46          DHDADI=0.0, TWISTA=0.0,
47          TYPE=1.0$
48 *****
49 *      Asymmetrical Control Deflection Parameters : Ailerons *
50 *****
51 $ASYFLP STYPE=4.0, NDELTA=9.0,
52          SPANFI=0.483, SPANFO=3.948, CHRDFI=0.508, CHRDFO=0.274,
53          DELTAL(1)=-20.0, -15.0, -10.0, -5.0, 0.0, 5.0, 10.0, 15.0, 20.0,
54          DELTAR(1)=20.0, 15.0, 10.0, 5.0, 0.0, -5.0, -10.0, -15.0, -20.0$
55 *****
56 * Wing Sectional Characteristics Parameters *
57 *****
58 NACA W 4 2414
59 SAVE
60 NEXT CASE
61
```



```

62 CASEID MQ-1 Predator: Tail, Fin
63 *****
64 *      Vertical Fin Sectional Characteristics *
65 *****
66 NACA F 4 0009
67 $VFPLNF CHRDR=0.711, CHRDTP=0.711,
68     SSPN=1.938, SSPNE=1.170,
69     TYPE=1.0$
70 *****
71 *      Inverted V Tail Sectional Characteristics *
72 *****
73 NACA V 4 0009
74 $VTPLNF CHRDR=0.711, CHRDTP=0.711,
75     SSPN=2.304, SSPNE=2.100,
76     TYPE=1.0$
77 SAVE
78 NEXT CASE
79
80 *****
81 *      Propeller Engine Characteristics *
82 *****
83 CASEID MQ-1 Predator: Engine eng=eng_Rotax914 prop=prop_Rtax914
84 $PROPWR NENGSP=1.0,
85     PHALOC=8.247, YP=0.0, PHVLOC=0.0,
86     NOPBPE=2.0, PRPRAD=0.9144, ENGFCT=0.85, AIETLP=0.0,
87     THSTCP=0.0,$
88 SAVE
89 NEXT CASE
90
91 CASEID MQ-1 Predator: Second Power Setting
92 $PROPWR THSTCP=0.60$
93 SAVE
94 NEXT CASE
95
96 CASEID MQ-1 Predator: Third Power Setting
97 $PROPWR THSTCP=0.65$
98 SAVE
99 NEXT CASE
100
101
102 CASEID MQ-1 Predator: Fourth Power Setting
103 $PROPWR THSTCP=0.70$
104 SAVE
105 NEXT CASE
106
107 CASEID MQ-1 Predator: Fifth Power Setting
108 $PROPWR THSTCP=0.75$
109 SAVE
110 NEXT CASE

```

Listing B.2: XML Engine File

```
1 <!--
2 File:      Rotax_914F.xml
3 Author:    Aero-Matic v 0.83
4 Inputs:
5 name:      Rotax_914F
6 type:      piston
7 power:     115.326 hp
8 augmented? no
9 injected?  no
10 -->
11
12 <piston_engine name="Rotax_914F">
13   <minmp unit="INHG">      34.0 </minmp>
14   <maxmp unit="INHG">      35.4 </maxmp>
15   <displacement unit="IN3"> 73.91 </displacement>
16   <maxhp>      115.33 </maxhp>
17   <cycles>      4.0 </cycles>
18   <idlerpm>     1400.0 </idlerpm>
19   <maxrpm>      5800.0 </maxrpm>
20   <sparkfaildrop> 0.1 </sparkfaildrop>
21   <volumetric-efficiency> 0.85 </volumetric-efficiency>
22   <man-press-lag> 0.1 </man-press-lag>
23   <static-friction unit="HP"> 0.58 </static-friction>
24   <starter-torque> 78.70 </starter-torque>
25   <starter-rpm> 3000 </starter-rpm>
26   <stroke unit="IN"> 2.4 </stroke>
27   <bore unit="IN"> 3.13 </bore>
28   <cylinders>    4.0 </cylinders>
29   <compression-ratio> 9.0 </compression-ratio>
30 </piston_engine>
```

Listing B.3: XML Propeller File

```

1  <!-- Generated by Aero-Matic v 1.1
2  Inputs:
3  horsepower: 115.326
4  pitch: fixed
5  max engine rpm: 5800
6  prop diameter (ft): 6
7  Outputs:
8  max prop rpm: 3127.17
9  gear ratio: 1.85
10 Cp0: 0.024228
11 Ct0: 0.033920
12 static thrust (lbs): 304.23
13 -->
14
15 <propeller version="1.01" name="prop">
16   <ixx> 2.64 </ixx>
17   <diameter unit="IN"> 72.0 </diameter>
18   <numblades> 2 </numblades>
19   <gearratio> 2.43 </gearratio>
20   <cp_factor> 1.00 </cp_factor>
21   <ct_factor> 1.00 </ct_factor>
22
23   <table name="C_THRUST" type="internal">
24     <tableData>
25       0.0 0.0370
26       0.1 0.0354
27       0.2 0.0339
28       0.3 0.0312
29       0.4 0.0280
30       0.5 0.0247
31       0.6 0.0200
32       0.7 0.0152
33       0.8 0.0082
34       1.0 -0.0028
35       1.2 -0.0146
36       1.4 -0.0262
37     </tableData>
38   </table>
39
40   <table name="C_POWER" type="internal">
41     <tableData>
42       0.0 0.0248
43       0.1 0.0248
44       0.2 0.0242
45       0.3 0.0236
46       0.4 0.0223
47       0.5 0.0204
48       0.6 0.0188
49       0.7 0.0157
50       0.8 0.0129
51       1.0 0.0046
52       1.2 -0.0073
53       1.4 -0.0221
54       1.6 -0.0375
55     </tableData>
56   </table>
57
58   <!-- thrust effects of helical tip Mach -->
59   <table name="CT_MACH" type="internal">
60     <tableData>
61       0.85 1.0
62       1.05 0.8
63     </tableData>

```

```

64     </table>
65
66
67     <!-- power-required effects of helical tip Mach -->
68     <table name="CP_MACH" type="internal">
69         <tableData>
70             0.85    1.0
71             1.05    1.8
72             2.00    1.4
73         </tableData>
74     </table>
75 </propeller>

```

B.2 Athena Vortex Lattice Data

Listing B.4: AVL Input File

```

1 *****
2 # AVL dataset for MQ-1 Predator *
3 *****
4 MQ-1 Predator
5 #Mach
6 0.108
7 #IYsym      IZsym      Zsym
8 0           0           0
9 #Sref       Cref       Bref
10 12.84       1.10       16.84
11
12 *****
13 # AVL Axes: *
14 # +X      downstream *
15 # +Y      out right wing *
16 # +Z      up *
17 *****
18 #Xref       Yref       Zref
19 4.04        0.00       3.00
20
21 *****
22 # Surfaces *
23 *****
24 #=====body=====
25 #BODY
26 #Fuse pod
27 # 14 1
28 #BFILE
29 #fuseMQ1.dat
30
31
32 #=====wing=====
33 SURFACE
34 wing
35 #Nchord      Cspace      Nspan      Sspace
36 10           1.0         50         1.0
37 SCALE
38 #sX          sY          sZ
39 1.0          1.0         1.0
40 TRANSLATE
41 #dX          dY          dZ
42 3.769        0.0         -0.2249
43 ANGLE
44 #Ainc
45 0
46
47 #=====wing section 1=====
48 SECTION
49 #Xle         Yle         Zle         Chord      Angle
50 0.3666       -8.42       0.0         0.40       0.0
51 NACA
52 #Airfoil definition
53 2414
54
55 #=====wing section 2=====
56 SECTION
57 #Xle         Yle         Zle         Chord      Angle
58 0.1719       -3.948       0.0         0.7894     0.0
59 NACA
60 #Airfoil definition
61 2414

```

```

62 CONTROL
63 #label      gain      Xhinge    Xhvec      Yhvec      Zhvec      SgnDup
64 aileron     1.0       0.653    0.0        0.0        0.0        1
65
66 #=====wing section 3=====
67 SECTION
68 #Xle        Yle        Zle        Chord      Angle
69 0.0210      -0.483    0.0       1.0912     0.0
70 NACA
71 #Airfoil definition
72 2414
73 CONTROL
74 #label      gain      Xhinge    Xhvec      Yhvec      Zhvec      SgnDup
75 aileron     1.0       0.534    0.0        0.0        0.0        1
76
77 #=====wing section 4=====
78 SECTION
79 #Xle        Yle        Zle        Chord      Angle
80 0.0         0.0       0.0       1.13       0.0
81 NACA
82 #Airfoil definition
83 2414
84
85 #=====wing section 5=====
86 SECTION
87 #Xle        Yle        Zle        Chord      Angle
88 0.0210      0.483    0.0       1.0912     0.0
89 NACA
90 #Airfoil definition
91 2414
92 CONTROL
93 #label      gain      Xhinge    Xhvec      Yhvec      Zhvec      SgnDup
94 aileron     1.0       0.534    0.0        0.0        0.0       -1
95
96 #=====wing section 6=====
97 SECTION
98 #Xle        Yle        Zle        Chord      Angle
99 0.1719      3.948    0.0       0.7894     0.0
100 NACA
101 #Airfoil definition
102 2414
103 CONTROL
104 #label      gain      Xhinge    Xhvec      Yhvec      Zhvec      SgnDup
105 aileron     1.0       0.653    0.0        0.0        0.0       -1
106
107 #=====wing section 7=====
108 SECTION
109 #Xle        Yle        Zle        Chord      Angle
110 0.3666      8.42     0.0       0.40       0.0
111 NACA
112 #Airfoil definition
113 2414
114
115 #=====V Tail 1=====
116 SURFACE
117 V tail 1
118 #Nchord     Cspace     Nspan     Sspace
119 12          2.0        15        1.0
120 SCALE
121 #sX         sY         sZ
122 1.0         1.0        1.0
123 TRANSLATE
124 #dX         dY         dZ
125 6.891       0.000     -0.200
126 ANGLE
127 #Ainc
128 0
129

```

```

130
131 #=====V Tail 1 section 1=====
132 SECTION
133 #Xle      Yle      Zle      Chord      Angle
134 0.0      -1.970    -1.379    0.711    -35.0
135 CONTROL
136 #label    gain      Xhinge    Xhvec     Yhvec     Zhvec     SgnDup
137 elevator  1.0        0.50     0.0       0.0       0.0       1
138
139 #=====V Tail 1 section 2=====
140 SECTION
141 #Xle      Yle      Zle      Chord      Angle
142 0.0      -0.305     0.0       0.711    -35.0000
143 CONTROL
144 #label    gain      Xhinge    Xhvec     Yhvec     Zhvec     SgnDup
145 elevator  1.0        0.50     0.0       0.0       0.0       1
146
147 #=====V TAIL 2=====
148 SURFACE
149 V tail 2
150 #Nchord    Cspace    Nspan      Sspace
151 12         2.0       15         1.0
152 SCALE
153 #sX        sY        sZ
154 1.0        1.0       1.0
155 TRANSLATE
156 #dX        dY        dZ
157 6.891     0.0       -0.20
158 ANGLE
159 #Ainc
160 0
161
162
163 #=====V Tail 2 section 1=====
164 SECTION
165 #Xle      Yle      Zle      Chord      Angle
166 0.0      1.970     -1.379    0.711    -35.0
167 CONTROL
168 #label    gain      Xhinge    Xhvec     Yhvec     Zhvec     SgnDup
169 elevator  1.0        0.50     0.0       0.0       0.0       1
170
171 #=====V Tail 2 section 2=====
172 SECTION
173 #Xle      Yle      Zle      Chord      Angle
174 0.0      0.305     0.0       0.711    -35.0
175 CONTROL
176 #label    gain      Xhinge    Xhvec     Yhvec     Zhvec     SgnDup
177 elevator  1.0        0.50     0.0       0.0       0.0       1
178
179 #=====vertical=====
180 SURFACE
181 vertical
182 #Nchord    Cspace    Nspan      Sspace
183 12         2.0       32         0.0
184 SCALE
185 #sX        sY        sZ
186 1.0        1.0       1.0
187 TRANSLATE
188 #dX        dY        dZ
189 6.891     0.0       -0.200
190 ANGLE
191 #Ainc
192 0
193
194 #=====vertical section 1=====
195 SECTION
196 #Xle      Yle      Zle      Chord      Angle
197 0.0      0.0       0.0       0.711    0.0

```

```

198 CONTROL
199 #label      gain      Xhinge      Xhvec      Yhvec      Zhvec      SgnDup
200 rudder      1.0        0.30       0.0        0.0        0.0        1
201
202 #=====vertical section 2=====
203 SECTION
204 #Xle        Yle        Zle        Chord      Angle
205 0.0         0.0        -1.170     0.711      0.0
206 CONTROL
207 #label      gain      Xhinge      Xhvec      Yhvec      Zhvec      SgnDup
208 rudder      1.0        0.30       0.0        0.0        0.0        1

```


Listing B.5: MQ-1 Predator Fuselage Coordinates File

1	Predator	fuselage
2	8.2467	0.0000
3	8.1443	0.1349
4	7.4194	0.2878
5	3.4965	0.2878
6	3.0701	0.2878
7	1.4924	0.4677
8	1.1939	0.4677
9	0.9381	0.3958
10	0.6823	0.2698
11	0.3838	0.0630
12	0.0000	-0.2698
13	0.3838	-0.5846
14	0.6823	-0.5846
15	0.9381	-0.5846
16	1.1939	-0.5846
17	1.4924	-0.5846
18	3.0701	-0.3418
19	3.4965	-0.3148
20	7.4194	-0.3148
21	8.1443	-0.1349

Listing B.6: Mass Properties File

```

1  # MQ-1 PREDATOR
2  # Mass & Inertia breakdown
3  #
4  # xyz is location of items own CG
5  # Ixx.. are items inertia about items own CG
6  #
7  # x back
8  # y right
9  # z up
10 #
11 # x,y,z system here must have origin
12 # at same location as AVL input file
13 #
14
15
16 Lunit = 1.0 m
17 Munit = 1.0 kg
18 Tunit = 1.0 s
19
20 g = 9.81
21 rho = 1.027
22
23 #
24 # mass    x      y      z      Ixx      Iyy      Izz      [ Ixy  Ixz  Iyz ]
25 *    1.0    1.0    1.0    1.0      1.0      1.0      1.0      1.0  1.0  1.0
26 +    0.0   -4.0178 0.0    0.0      0.0      0.0      0.0      0.0  0.0  0.0
27      800.0    0.1210 0.0    0.0     6800    11500    18500      ! aircraft

```

Listing B.7: NACA 2414 Airfoil Coordinates File

1	NACA 2414	
2	1.00000	0.00147
3	0.99739	0.00210
4	0.98929	0.00396
5	0.97587	0.00700
6	0.95729	0.01112
7	0.93372	0.01620
8	0.90542	0.02207
9	0.87267	0.02857
10	0.83582	0.03552
11	0.79527	0.04274
12	0.75143	0.05004
13	0.70480	0.05723
14	0.65586	0.06412
15	0.60515	0.07053
16	0.55324	0.07629
17	0.50069	0.08120
18	0.44808	0.08512
19	0.39598	0.08787
20	0.34454	0.08913
21	0.29482	0.08866
22	0.24740	0.08645
23	0.20285	0.08255
24	0.16169	0.07707
25	0.12440	0.07014
26	0.09141	0.06198
27	0.06310	0.05281
28	0.03977	0.04289
29	0.02165	0.03245
30	0.00892	0.02171
31	0.00169	0.01085
32	0.00000	0.00000
33	0.00379	-0.01031
34	0.01293	-0.01956
35	0.02730	-0.02770
36	0.04669	-0.03471
37	0.07087	-0.04054
38	0.09957	-0.04516
39	0.13246	-0.04858
40	0.16918	-0.05082
41	0.20937	-0.05195
42	0.25260	-0.05208
43	0.29844	-0.05133
44	0.34644	-0.04987
45	0.39611	-0.04787
46	0.44739	-0.04537
47	0.49931	-0.04232
48	0.55129	-0.03886
49	0.60276	-0.03516
50	0.65316	-0.03132
51	0.70194	-0.02745
52	0.74857	-0.02365
53	0.79252	-0.01998
54	0.83331	-0.01650
55	0.87048	-0.01328
56	0.90360	-0.01035
57	0.93230	-0.00776
58	0.95626	-0.00557
59	0.97518	-0.00381
60	0.98886	-0.00252
61	0.99713	-0.00173
62	1.00000	-0.00147

Appendix C

SIMULATION PARAMETER FILES

Table C.1: Simulation scenario initial parameters.

Parameter	Value
Julian date	2454680.0
Start coordinates	34.726N, 120.577W
Target coordinates	34.804N, 120.603W
Axial speed	36 m/s
Altitude	1800 m
Angle of attack, α	0 deg
Sideslip angle, β	0 deg
Roll, ϕ	0 deg
Pitch, θ	0 deg
Yaw, ψ	0 deg
Camera horizontal FOV	360 m
Camera vertical FOV	360 m
Pheromone deposit	20
Pheromone propagation	0.75
Pheromone evaporation	0.03
Pheromone threshold	1e-300
Pheromone update	10
Total time	800.0 sec
Time-step	0.1 sec

C.1 Stability and Control Derivatives

$$\mathbf{A} = \begin{bmatrix} -0.0780 & 0.0578 & -0.3073 & -9.8067 & -0.0235 & 0.1071 & 0.0453 & 0 & 0 & 0 & 0.0144 \\ -0.1109 & -1.8798 & 33.9237 & 0 & 0.0100 & 0.0005 & 0.2491 & 0 & 0 & 0 & 0 \\ -0.0257 & -0.0136 & -0.2240 & 0 & -0.0043 & 0.0345 & 0.0603 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.1747 & -0.0040 & 0.6763 & 0 & -0.1226 & 0.4794 & -35.6559 & 9.8067 & 0 & 0 & 0 \\ -0.1114 & -0.0057 & -0.4548 & 0 & 0.0708 & -3.4288 & 0.0514 & 0 & 0 & 0 & 0 \\ -0.0225 & -0.0013 & -0.0910 & 0 & 0.0095 & -0.0714 & -0.0697 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 36 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \quad (\text{C.1})$$

$$\mathbf{B} = \begin{bmatrix} 4.73E-03 & 5.95E-02 & 2.54E-03 & 0 \\ -0.3673 & 1.43E-03 & -3.98E-03 & 0 \\ -5.61E-05 & 1.61E-02 & -1.21E-04 & 0 \\ 0 & 0 & 0 & 0 \\ -4.62E-04 & -6.61E-02 & 2.12E-02 & 0 \\ -2.92E-04 & 4.19E-02 & -8.63E-03 & 0 \\ -2.57E-04 & 8.91E-03 & -2.77E-03 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{C.2})$$

C.2 Set-Point Control Matrices

$$\mathbf{F} = \begin{bmatrix} 0 & 0.500000000000000 & 0 & 0 \\ 0 & 0.029194324007005 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0.000810953444639 & 0 & 0 \\ 0 & 0.235589577880307 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -0.006271580615290 & 0 & 0 \\ 0 & -0.022297078091798 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.074556723460457 & 0 & 0 \end{bmatrix} \quad (\text{C.3})$$

$$\mathbf{K} = \begin{bmatrix} 7.6569 & -18.8079 & 29.1745 & 881.4351 & -0.0351 & -0.4843 & 2.6611 & -1.5458 & 182.8348 & 99.6041 & -0.0156 \\ 345.3983 & 1.9796 & 5.5087 & -190.8778 & -0.4164 & 0.5056 & 8.6668 & -3.7181 & 3502.0197 & -5.5551 & 0.2175 \\ 29.2709 & 0.2286 & -33.3854 & -97.9460 & 0.7599 & 23.4717 & -146.5891 & 71.5036 & 170.8522 & 1.1848 & 0.0779 \\ 614.3180 & 9.2832 & -1929.8570 & -3330.0058 & -2.1829 & 8.6991 & -86.3439 & 24.5245 & 944.4795 & 26.5166 & 3.1716 \end{bmatrix} \quad (\text{C.4})$$

$$\mathbf{N} = \begin{bmatrix} 0 & -0.292902136516323 & 0 & 0 \\ 0 & 0.906949080372875 & 0 & 0 \\ 0 & -0.164812385290060 & 0 & 0 \\ 0 & -0.074556723460457 & 0 & 0 \end{bmatrix} \quad (\text{C.5})$$

C.3 Simulation Batch File

Listing C.1: Batch File to Run HSF

```
1 echo off
2
3 rem run through the different size swarms
4 for /l %%a in (1,1,4) do (
5
6     rem change the visited pheromone update value
7     for /l %%b in (2,8,18) do (
8
9         rem specifies the visit pheromone deposit
10        for /l %%c in (2,8,18) do (
11
12            rem specifies the visit pheromone threshold before disappearing
13            for /l %%d in (2,8,18) do (
14
15                rem create XML file
16                cd C:\<path_to_python_script>
17                python <script_name>.py %%a %%b %%c 1e-%%d
18
19                rem change directory and run HSF .exe
20                cd C:\<path_to_HSF>\Horizon\Horizon\bin\Debug
21                Horizon.exe
22
23                rem rename and move HSF output data for every asset
24                ren "C:\<path_to_HSF>\Horizon\*dynamicStateData.csv"~
25                "????????????????????_up%%b_dep%%c_thr%%d.csv"
26                move /y "C:\<path_to_HSF>\Horizon\*_dynamicStateData*.csv"~
27                "<new_directory_path>"
28
29                rem move subsystem data
30                ren "C:\HorizonLog\Scratch\*_fuel.csv"~
31                "????????????_up%%b_dep%%c_thr%%d.csv"
32                ren "C:\HorizonLog\Scratch\*_targetfound.csv"~
33                "????????????????????_up%%b_dep%%c_thr%%d.csv"
34                ren "C:\HorizonLog\Scratch\*_targetdistance.csv"~
35                "????????????????????_up%%b_dep%%c_thr%%d.csv"
36                move "C:\HorizonLog\Scratch\*.csv" "<new_directory_path>"
37            )
38        )
39    )
40 )
```


Appendix D

SIMULATION OUTPUT DATA

D.1 Swarm: 2 Assets

Table D.1: 2 asset swarm output data.

Update	Deposit	Asset	Parameter	Thrsh 1e-02	Thrsh 1e-10	Thrsh 1e-18
2	2	1	Max distance	1310.21 m	945.92 m	1141.05 m
2	2	1	Mean distance	698.44 m	526.12 m	654.39 m
2	2	1	Fuel consumed	0.5568 gal	0.5361 gal	0.5382 gal
2	2	2	Max distance	916.14 m	912.02 m	975.49 m
2	2	2	Mean distance	453.62 m	456.41 m	418.59 m
2	2	2	Fuel consumed	0.5430 gal	0.5409 gal	0.5409 gal
2	10	1	Max distance	1042.89 m	942.54 m	942.54 m
2	10	1	Mean distance	471.08 m	440.35 m	547.61 m
2	10	1	Fuel consumed	0.5518 gal	0.5399 gal	0.5383 gal
2	10	2	Max distance	799.55 m	913.51 m	772.10 m
2	10	2	Mean distance	377.78 m	500.65 m	438.28 m
2	10	2	Fuel consumed	0.5480 gal	0.5390 gal	0.5386 gal
2	18	1	Max distance	1062.56 m	942.54 m	942.54 m
2	18	1	Mean distance	458.50 m	440.35 m	547.61 m
2	18	1	Fuel consumed	0.5609 gal	0.5399 gal	0.5383 gal
2	18	2	Max distance	729.89 m	913.51 m	772.10 m
2	18	2	Mean distance	360.48 m	500.65 m	438.28 m
2	18	2	Fuel consumed	0.5499 gal	0.5390 gal	0.5386 gal

10	2	1	Max distance	954.85 m	1737.64 m	1990.00 m
10	2	1	Mean distance	474.83 m	999.80 m	959.81 m
10	2	1	Fuel consumed	0.5399 gal	0.5323 gal	0.5357 gal
10	2	2	Max distance	1013.17 m	1492.78 m	1508.16 m
10	2	2	Mean distance	480.00 m	765.41 m	844.27 m
10	2	2	Fuel consumed	0.5390 gal	0.5330 gal	0.5319 gal
10	10	1	Max distance	1167.99 m	2219.30 m	2338.17 m
10	10	1	Mean distance	610.23 m	1274.89 m	1238.25 m
10	10	1	Fuel consumed	0.5361 gal	0.5316 gal	0.5329 gal
10	10	2	Max distance	1136.32 m	1871.92 m	1938.05 m
10	10	2	Mean distance	460.05 m	953.49 m	1212.45 m
10	10	2	Fuel consumed	0.5401 gal	0.5308 gal	0.5301 gal
10	18	1	Max distance	1072.02 m	1937.57 m	1955.30 m
10	18	1	Mean distance	537.78 m	971.19 m	1001.05 m
10	18	1	Fuel consumed	0.5359 gal	0.5288 gal	0.5311 gal
10	18	2	Max distance	1046.11 m	1752.40 m	2037.64 m
10	18	2	Mean distance	593.76 m	820.63 m	897.01 m
10	18	2	Fuel consumed	0.5341 gal	0.5349 gal	0.5337 gal
18	2	1	Max distance	942.54 m	1926.65 m	2445.34 m
18	2	1	Mean distance	492.01 m	834.15 m	1346.74 m
18	2	1	Fuel consumed	0.5351 gal	0.5321 gal	0.5308 gal
18	2	2	Max distance	1243.47 m	1788.08 m	2142.26 m
18	2	2	Mean distance	718.32 m	1136.56 m	1324.97 m
18	2	2	Fuel consumed	0.5369 gal	0.5357 gal	0.5315 gal
18	10	1	Max distance	1510.98 m	2165.83 m	2110.95 m
18	10	1	Mean distance	699.88 m	1170.97 m	1319.35 m

18	10	1	Fuel consumed	0.5335 gal	0.5309 gal	0.5294 gal
18	10	2	Max distance	1439.08 m	1742.82 m	2094.30 m
18	10	2	Mean distance	701.29 m	1070.71 m	1252.70 m
18	10	2	Fuel consumed	0.5331 gal	0.5307 gal	0.5300 gal
18	18	1	Max distance	1654.25 m	2007.87 m	1956.30 m
18	18	1	Mean distance	746.78 m	1218.74 m	1017.07 m
18	18	1	Fuel consumed	0.5319 gal	0.5289 gal	0.5328 gal
18	18	2	Max distance	1439.08 m	2072.17 m	1652.48 m
18	18	2	Mean distance	767.60 m	1178.01 m	1017.76 m
18	18	2	Fuel consumed	0.5351 gal	0.5297 gal	0.5291 gal

D.2 Swarm: 3 Assets

Table D.2: 3 asset swarm output data.

Update	Deposit	Asset	Parameter	Thrsh 1e-02	Thrsh 1e-10	Thrsh 1e-18
2	2	1	Max distance	781.46 m	1176.60 m	1333.80 m
2	2	1	Mean distance	394.18 m	631.08 m	644.00 m
2	2	1	Fuel consumed	0.5443 gal	0.5361 gal	0.5346 gal
2	2	2	Max distance	958.55 m	1301.43 m	1301.43 m
2	2	2	Mean distance	432.50 m	648.77 m	705.66 m
2	2	2	Fuel consumed	0.5381 gal	0.5333 gal	0.5355 gal
2	2	3	Max distance	887.00 m	1424.18 m	1265.34 m
2	2	3	Mean distance	445.00 m	688.58 m	639.44 m
2	2	3	Fuel consumed	0.5499 gal	0.5552 gal	0.5359 gal
2	10	1	Max distance	933.75 m	1080.78 m	1300.82 m
2	10	1	Mean distance	487.10 m	609.34 m	648.57 m
2	10	1	Fuel consumed	0.5428 gal	0.5348 gal	0.5348 gal
2	10	2	Max distance	1122.93 m	1301.43 m	1301.43 m
2	10	2	Mean distance	462.18 m	653.79 m	656.52 m
2	10	2	Fuel consumed	0.5528 gal	0.5334 gal	0.5348 gal
2	10	3	Max distance	1020.87 m	1265.34 m	1265.34 m
2	10	3	Mean distance	561.44 m	555.96 m	561.98 m
2	10	3	Fuel consumed	0.5597 gal	0.5461 gal	0.5352 gal
2	18	1	Max distance	916.89 m	1369.32 m	1080.78 m
2	18	1	Mean distance	447.04 m	677.79 m	585.12 m
2	18	1	Fuel consumed	0.5401 gal	0.5356 gal	0.5369 gal
2	18	2	Max distance	918.57 m	1690.18 m	1301.43 m

2	18	2	Mean distance	414.37 m	763.03 m	651.03 m
2	18	2	Fuel consumed	0.5500 gal	0.5392 gal	0.5358 gal
2	18	3	Max distance	1008.77 m	1405.26 m	1303.47 m
2	18	3	Mean distance	447.38 m	628.01 m	709.34 m
2	18	3	Fuel consumed	0.5420 gal	0.5415 gal	0.5381 gal
10	2	1	Max distance	1486.83 m	1903.48 m	1681.95 m
10	2	1	Mean distance	856.14 m	1112.29 m	837.57 m
10	2	1	Fuel consumed	0.5320 gal	0.5296 gal	0.5326 gal
10	2	2	Max distance	1301.43 m	1905.74 m	2583.56 m
10	2	2	Mean distance	582.35 m	940.96 m	1514.34 m
10	2	2	Fuel consumed	0.5342 gal	0.5300 gal	0.5277 gal
10	2	3	Max distance	1265.34 m	2043.41 m	2394.64 m
10	2	3	Mean distance	675.88 m	1208.49 m	1409.43 m
10	2	3	Fuel consumed	0.5359 gal	0.5302 gal	0.5309 gal
10	10	1	Max distance	1262.39 m	1960.43 m	2373.73 m
10	10	1	Mean distance	639.38 m	1061.18 m	1113.18 m
10	10	1	Fuel consumed	0.5352 gal	0.5318 gal	0.5270 gal
10	10	2	Max distance	1327.79 m	1672.29 m	2583.56 m
10	10	2	Mean distance	799.42 m	1027.87 m	1466.08 m
10	10	2	Fuel consumed	0.5321 gal	0.5303 gal	0.5271 gal
10	10	3	Max distance	1265.34 m	2043.41 m	2394.64 m
10	10	3	Mean distance	595.16 m	1135.62 m	1536.20 m
10	10	3	Fuel consumed	0.5407 gal	0.5301 gal	0.5298 gal
10	18	1	Max distance	1471.09 m	1954.59 m	1936.83 m
10	18	1	Mean distance	575.67 m	1068.66 m	888.19 m
10	18	1	Fuel consumed	0.5353 gal	0.5321 gal	0.5305 gal

10	18	2	Max distance	1618.33 m	1672.29 m	2583.56 m
10	18	2	Mean distance	794.29 m	1027.87 m	1620.69 m
10	18	2	Fuel consumed	0.5306 gal	0.5303 gal	0.5271 gal
10	18	3	Max distance	1296.99 m	2043.41 m	2394.64 m
10	18	3	Mean distance	750.08 m	1278.41 m	1430.76 m
10	18	3	Fuel consumed	0.5348 gal	0.5306 gal	0.5321 gal
18	2	1	Max distance	1391.57 m	2399.15 m	2587.89 m
18	2	1	Mean distance	594.07 m	1565.48 m	1663.72 m
18	2	1	Fuel consumed	0.5378 gal	0.5280 gal	0.5294 gal
18	2	2	Max distance	1778.72 m	2583.56 m	2583.56 m
18	2	2	Mean distance	955.41 m	1622.27 m	1678.57 m
18	2	2	Fuel consumed	0.5351 gal	0.5267 gal	0.5276 gal
18	2	3	Max distance	1469.06 m	2047.99 m	2539.50 m
18	2	3	Mean distance	899.00 m	1149.24 m	1600.99 m
18	2	3	Fuel consumed	0.5320 gal	0.5307 gal	0.5318 gal
18	10	1	Max distance	1662.46 m	2399.15 m	2587.89 m
18	10	1	Mean distance	1029.67 m	1565.48 m	1663.72 m
18	10	1	Fuel consumed	0.5315 gal	0.5280 gal	0.5294 gal
18	10	2	Max distance	1653.03 m	2583.56 m	2583.56 m
18	10	2	Mean distance	945.63 m	1622.27 m	1678.57 m
18	10	2	Fuel consumed	0.5285 gal	0.5267 gal	0.5276 gal
18	10	3	Max distance	1592.36 m	2047.99 m	2539.50 m
18	10	3	Mean distance	744.81 m	1149.24 m	1600.99 m
18	10	3	Fuel consumed	0.5327 gal	0.5307 gal	0.5318 gal
18	18	1	Max distance	1663.54 m	2399.15 m	2587.89 m
18	18	1	Mean distance	869.95 m	1565.48 m	1663.72 m

18	18	1	Fuel consumed	0.5319 gal	0.5280 gal	0.5294 gal
18	18	2	Max distance	1832.96 m	2583.56 m	2583.56 m
18	18	2	Mean distance	1151.99 m	1622.27 m	1678.57 m
18	18	2	Fuel consumed	0.5302 gal	0.5267 gal	0.5276 gal
18	18	3	Max distance	1545.40 m	2047.99 m	2539.50 m
18	18	3	Mean distance	915.46 m	1149.24 m	1600.99 m
18	18	3	Fuel consumed	0.5306 gal	0.5307 gal	0.5318 gal

D.3 Swarm: 4 Assets

Table D.3: 4 asset swarm output data.

Update	Deposit	Asset	Parameter	Thrsh 1e-02	Thrsh 1e-10	Thrsh 1e-18
2	2	1	Max distance	996.34 m	1393.03 m	1324.13 m
2	2	1	Mean distance	461.86 m	696.94 m	641.43 m
2	2	1	Fuel consumed	0.5546 gal	0.5344 gal	0.5360 gal
2	2	2	Max distance	1000.61 m	1329.38 m	1301.43 m
2	2	2	Mean distance	487.85 m	722.14 m	651.64 m
2	2	2	Fuel consumed	0.5584 gal	0.5369 gal	0.5333 gal
2	2	3	Max distance	992.74 m	1265.34 m	1461.42 m
2	2	3	Mean distance	460.45 m	590.05 m	843.75 m
2	2	3	Fuel consumed	0.5397 gal	0.5378 gal	0.5318 gal
2	2	4	Max distance	1037.08 m	1489.92 m	1486.90 m
2	2	4	Mean distance	403.24 m	603.97 m	626.47 m
2	2	4	Fuel consumed	0.5468 gal	0.5387 gal	0.5414 gal
2	10	1	Max distance	1284.07 m	1475.72 m	1378.04 m
2	10	1	Mean distance	644.12 m	736.42 m	708.16 m
2	10	1	Fuel consumed	0.5558 gal	0.5373 gal	0.5351 gal
2	10	2	Max distance	1290.78 m	1301.43 m	1511.37 m
2	10	2	Mean distance	651.11 m	678.30 m	777.49 m
2	10	2	Fuel consumed	0.5420 gal	0.5343 gal	0.5360 gal
2	10	3	Max distance	955.32 m	1265.34 m	1265.34 m
2	10	3	Mean distance	461.38 m	526.69 m	705.38 m
2	10	3	Fuel consumed	0.5396 gal	0.5358 gal	0.5375 gal
2	10	4	Max distance	1037.08 m	1486.90 m	1486.90 m

2	10	4	Mean distance	497.64 m	564.81 m	750.51 m
2	10	4	Fuel consumed	0.5443 gal	0.5395 gal	0.5410 gal
2	18	1	Max distance	1278.34 m	1494.22 m	1494.22 m
2	18	1	Mean distance	564.82 m	630.35 m	614.07 m
2	18	1	Fuel consumed	0.5421 gal	0.5348 gal	0.5383 gal
2	18	2	Max distance	1177.91 m	1642.15 m	1642.15 m
2	18	2	Mean distance	461.93 m	796.57 m	805.35 m
2	18	2	Fuel consumed	0.5436 gal	0.5371 gal	0.5357 gal
2	18	3	Max distance	1031.09 m	1479.06 m	2062.04 m
2	18	3	Mean distance	465.21 m	760.17 m	845.71 m
2	18	3	Fuel consumed	0.5423 gal	0.5345 gal	0.5487 gal
2	18	4	Max distance	1103.18 m	1756.12 m	1507.19 m
2	18	4	Mean distance	492.02 m	809.02 m	910.81 m
2	18	4	Fuel consumed	0.5552 gal	0.5425 gal	0.5338 gal
10	2	1	Max distance	1494.22 m	2192.67 m	1797.94 m
10	2	1	Mean distance	681.13 m	1220.95 m	992.33 m
10	2	1	Fuel consumed	0.5367 gal	0.5335 gal	0.5314 gal
10	2	2	Max distance	1642.15 m	1642.15 m	2562.94 m
10	2	2	Mean distance	903.78 m	899.13 m	1296.60 m
10	2	2	Fuel consumed	0.5350 gal	0.5291 gal	0.5281 gal
10	2	3	Max distance	1265.34 m	2507.52 m	2837.62 m
10	2	3	Mean distance	632.83 m	1478.50 m	1816.98 m
10	2	3	Fuel consumed	0.5449 gal	0.5262 gal	0.5256 gal
10	2	4	Max distance	1486.90 m	2039.52 m	3256.41 m
10	2	4	Mean distance	744.44 m	1121.22 m	2001.42 m
10	2	4	Fuel consumed	0.5336 gal	0.5334 gal	0.5293 gal

10	10	1	Max distance	1494.22 m	2129.93 m	1797.94 m
10	10	1	Mean distance	784.60 m	1178.51 m	1047.04 m
10	10	1	Fuel consumed	0.5367 gal	0.5293 gal	0.5344 gal
10	10	2	Max distance	1642.15 m	2270.58 m	2414.37 m
10	10	2	Mean distance	575.17 m	1383.59 m	1365.31 m
10	10	2	Fuel consumed	0.5357 gal	0.5286 gal	0.5308 gal
10	10	3	Max distance	1490.05 m	1836.18 m	1830.18 m
10	10	3	Mean distance	801.47 m	1175.75 m	1011.59 m
10	10	3	Fuel consumed	0.5327 gal	0.5296 gal	0.5296 gal
10	10	4	Max distance	1486.90 m	1949.82 m	2637.85 m
10	10	4	Mean distance	813.74 m	1144.92 m	1675.35 m
10	10	4	Fuel consumed	0.5365 gal	0.5336 gal	0.5315 gal
10	18	1	Max distance	1822.66 m	2129.93 m	2338.07 m
10	18	1	Mean distance	1054.68 m	1182.97 m	1330.94 m
10	18	1	Fuel consumed	0.5354 gal	0.5312 gal	0.5331 gal
10	18	2	Max distance	1642.15 m	2270.58 m	1821.91 m
10	18	2	Mean distance	752.98 m	1383.59 m	948.22 m
10	18	2	Fuel consumed	0.5337 gal	0.5286 gal	0.5319 gal
10	18	3	Max distance	1548.73 m	1836.18 m	1966.03 m
10	18	3	Mean distance	915.83 m	1177.47 m	1144.05 m
10	18	3	Fuel consumed	0.5337 gal	0.5301 gal	0.5343 gal
10	18	4	Max distance	1486.90 m	1949.82 m	2416.52 m
10	18	4	Mean distance	622.18 m	1144.92 m	1752.25 m
10	18	4	Fuel consumed	0.5370 gal	0.5336 gal	0.5322 gal
18	2	1	Max distance	1494.22 m	2137.75 m	2137.75 m
18	2	1	Mean distance	829.50 m	1166.66 m	1136.84 m

18	2	1	Fuel consumed	0.5349 gal	0.5331 gal	0.5334 gal
18	2	2	Max distance	1944.15 m	2890.13 m	2954.36 m
18	2	2	Mean distance	1076.73 m	1423.03 m	1665.66 m
18	2	2	Fuel consumed	0.5296 gal	0.5283 gal	0.5280 gal
18	2	3	Max distance	1509.60 m	2551.40 m	2551.40 m
18	2	3	Mean distance	816.90 m	1691.01 m	1755.10 m
18	2	3	Fuel consumed	0.5310 gal	0.5286 gal	0.5296 gal
18	2	4	Max distance	1529.96 m	3261.59 m	2937.04 m
18	2	4	Mean distance	721.88 m	1965.02 m	1918.61 m
18	2	4	Fuel consumed	0.5411 gal	0.5299 gal	0.5299 gal
18	10	1	Max distance	1494.22 m	3426.12 m	2965.99 m
18	10	1	Mean distance	869.89 m	1811.37 m	1853.64 m
18	10	1	Fuel consumed	0.5315 gal	0.5319 gal	0.5312 gal
18	10	2	Max distance	1832.33 m	3651.97 m	1821.91 m
18	10	2	Mean distance	967.13 m	2032.00 m	1011.34 m
18	10	2	Fuel consumed	0.5307 gal	0.5270 gal	0.5328 gal
18	10	3	Max distance	1529.18 m	3458.23 m	2551.40 m
18	10	3	Mean distance	954.94 m	1996.87 m	1755.10 m
18	10	3	Fuel consumed	0.5338 gal	0.5310 gal	0.5296 gal
18	10	4	Max distance	1842.30 m	3126.17 m	2937.04 m
18	10	4	Mean distance	1074.93 m	1925.05 m	1918.61 m
18	10	4	Fuel consumed	0.5331 gal	0.5322 gal	0.5299 gal
18	18	1	Max distance	1986.78 m	3499.66 m	2137.75 m
18	18	1	Mean distance	984.19 m	1831.63 m	1209.24 m
18	18	1	Fuel consumed	0.5303 gal	0.5312 gal	0.5345 gal
18	18	2	Max distance	1821.91 m	2871.57 m	2871.57 m

18	18	2	Mean distance	1151.33 m	1901.47 m	1901.47 m
18	18	2	Fuel consumed	0.5324 gal	0.5265 gal	0.5265 gal
18	18	3	Max distance	1862.72 m	3194.84 m	3194.84 m
18	18	3	Mean distance	1212.56 m	1981.83 m	1981.83 m
18	18	3	Fuel consumed	0.5303 gal	0.5300 gal	0.5300 gal
18	18	4	Max distance	1615.99 m	3384.02 m	3130.92 m
18	18	4	Mean distance	816.05 m	1994.47 m	1923.34 m
18	18	4	Fuel consumed	0.5343 gal	0.5340 gal	0.5345 gal