System Requirements Specification

for

Modeling Unmanned Aerial Swarms Using Unreal Engine and AirSim Simulator

Version 2.3.1 approved

Prepared by Dillon Mead

Embry-Riddle Aeronautical University EECS Department

08 APR 2022

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Dillon Mead	13/09/21	Initial document creation.	1.0.0
Dillon Mead	27/09/21	First Semester first version edit.	1.1.0
Dillon Mead	26/10/21	First Semester second version edit.	1.2.0
John Mueller	26/10/21	Added Stimulus/Response in section 4.	1.2.1
Dillon Mead	29/11/21	First Semester Third version edit.	1.3.0
Dillon Mead	30/11/21	Edits sections 2 and 4. Adds highlight convention.	1.3.1

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

Dillon Mead	04/02/22	Second Semester first version.	2.1.0
Dillon Mead	06/03/22	Second Semester second version.	2.2.0
Dillon Mead	05/04/22	Adds figures for visual aid.	2.3.0
Dillon Mead	08/04/22	Second Semester final version.	2.3.1

1. Introduction

1.1 Purpose

2nd Semester:
Current regulations restrict operation of Unmanned Aerial Vehicles (UAV). These restrictions include manual control by a human pilot within a specific radius and within Visual Line of Sight (VLOS). However, Government agencies and private enterprise wish to use autonomous UAV beyond VLOS. This isn't publicly accepted because of the potential risks involved. Autonomous systems, especially aircraft, operating over long distances without immediate supervision may not be considered safe and responsible.
The purpose of this project, and this SRS, is to define the requirements for a simulated unmanned aerial swarm in Microsoft's AirSim simulator [1]. This product is also dependent on Unreal Engine 4 [3] for the environment and physics. The intent is to create a simulation model that can be utilized to test autonomous algorithms and behaviors to verify and validate the safety of autonomous UAV in a relatively inexpensive manner.

1.2 Document Conventions

Text highlighted in yellow are portions of the product not currently developed. These features will be explored during future development. Features completed during the **2nd Semester** will be identified by this label and by separate paragraphs.

1.3 Intended Audience and Reading Suggestions

This document is intended for developers, project managers, testers, and document writers.
The overview of this document is:

Section 1: Discusses the purpose and scope of the product, and explains the conventions within the document.
Section 2: Explains what is expected of the product overall, why the product is developed, lists the reference of other supporting documents for this product, describes the operating environment, and finally any assumptions, dependencies, or constraints.
Section 3: Describes the requirements for any interfaces the product relies on to operate.
Section 4: Describes the requirements and expected behavior of the simulation model.
Section 5: Describes other requirements necessary for the product operation, such as performance, safety, and software quality.
Appendix A: Is a glossary of technical terminology used within this SRS.
Appendix B: Contains any models or figures that illustrate requirements listed in this SRS and are also not included in the other supporting documents of this product.

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

1.4 Product Scope

This system simulates an unmanned aerial swarm. The aerial swarm will behave in a unified, cohesive manner. The general mission for the aerial swarm will be to collect data from a three-dimensional environment.

2nd Semester:
This system will employ a collision avoidance algorithm that integrates with the aerial swarm pathing method and includes object detection capabilities. The collision avoidance will activate for both stationary and moving objects.

1.5 References

[1] Microsoft Corporation. "AirSim." Version 1.6.0. August 2021. https://github.com/Microsoft/AirSim.
[2] Microsoft Corporation. "AirSim FAQ." https://microsoft.github.io/AirSim/faq/#what-computer-do-you-need.
[3] Epic Games, Inc. "Unreal Engine." Version 4.27. August 2021. https://www.unrealengine.com/en-US/.
[4] Aerial Swarm Simulators. "Modeling Unmanned Aerial Swarms Using Unreal Engine and AirSim Simulator System Design Document." Version 3.0. November 30, 2021. https://github.com/mead-d/Modeling-Unmanned-Aerial-Swarms-Using-Unreal-Game-Engine-and-AirSim-Simulator/tree/main/Deliverables.
[5] CFI Notebook. "Rules and Regulations/ Right-Of-Way." CFI Notebook. https://www.cfinotebook.net/notebook/rules-and-regulations/right-of-way (accessed Apr 06, 2022).

2. Overall Description

2.1 Product Perspective

This system is a new product inspired by earlier research from Professor Ilhan Akbas. The original idea was to use aerial wireless sensor and actor networks (aerial swarm) to perform observation tasks while mitigating risk to the user. For example, the user could be a geologist and the observation task is the smoke plume from a volcano. It is easier to gather data with an aerial swarm and minimize potential injury or loss of life.

2.2 Product Functions

- Multiple Unmanned Aerial Vehicles operate as unified, cohesive swarm.
- Each UAV maintains communication with the swarm.
- The aerial swarm maintains communication with the ground control station.
- The aerial swarm operates and maneuvers with a formation specified by the number of UAV.
- The ground station may transmit changed mission orders to the swarm.
- The aerial swarm and/ or individual UAV detect obstacles along mission path.
- The aerial swarm and/ or individual UAV evaluate possible collisions.
- The aerial swarm and/ or individual UAV avoids collisions using the FAA Right-of-Way rules.

2.3 User Classes and Characteristics

There is only one user class for this system. The user must have knowledge of computers and programming. The user must know how to operate in the AirSim environment. This product is developed with python 3.5 – 3.8, therefore the user must also need to understand the python language and how to operate in the python environment. Developers and Testers have the same user rights and capabilities.

2.4 Operating Environment

This system must use Microsoft's AirSim Simulator [1]. Microsoft's AirSim may be used on any computer using either Windows, MacOS, or Linux. There is no explicit hardware requirement when using the basic "blocks" environment [2]. A minimum of 4GB VRAM (GPU RAM) is required when using a more detailed environment. The current minimum software necessary for Microsoft's AirSim is Visual Studio 2019 and Unreal Engine 4.26.

2.5 Design and Implementation Constraints

The supporting software, Unreal Engine 4.27 [3], Microsoft's AirSim, Visual Studio 2019 (VS19), and additional tools or extensions must be installed and built in a specific manner. Microsoft provides a list of instructions for installation and building at the following location: https://microsoft.github.io/AirSim/build_windows/. Other difficulties may occur during installation that are not within Microsoft's directions. Chiefly, administrative rights are necessary throughout the installation and build process on the installation computer. C++ tools are required in VS19 and it is important to include the python tools when necessary. This system uses those python libraries available through AirSim to implement the product.

2.6 User Documentation

The most recent version of the System Design Document, System Requirements Specification, and Test Plan may be found within the Aerial Swarm Simulator team's GitHub repository within the deliverables folder: https://github.com/mead-d/Modeling-Unmanned-Aerial-Swarms-Using-Unreal-Game-Engine-and-AirSim-Simulator/tree/main/Deliverables.
The only exception is the AirSim Simulator documentation that may be found at: https://microsoft.github.io/AirSim/.

2.7 Assumptions and Dependencies

Assumptions:
We assume that Unreal Engine 4 and Microsoft's AirSim Simulator will remain open source.
The aerial swarm must have at least three UAV to properly conduct a given mission.
The operating environment is assumed to be clear of an overwhelming amount of obstacles.
There are no adversarial actors searching for the aerial swarm.

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

Dependencies:
This system is currently only usable within Microsoft's AirSim environment. AirSim is itself dependent on Unreal Engine 4 and Visual Studio 2019. Any changes to Unreal Engine 4, Visual Studio 2019, or Microsoft's AirSim have the potential to cause errors and irregularities within the Aerial Swarm Simulator system.

3. External Interface Requirements

3.1 User Interfaces

[Req 1] The Aerial Swarm Simulator system shall be modelled and simulated in Microsoft's AirSim Simulator.
[Req 2] Visual Studio 2019 shall be used to edit files and environment variables.
[Req 3] The User shall implement mission scenarios by executing the appropriate script through Visual Studio 2019 within the AirSim environment.
[Req 4] System data shall be display to the user through the python environment terminal.

3.2 Hardware Interfaces

This system has no hardware interface requirements.

3.3 Software Interfaces

Microsoft's AirSim Simulator interfaces with Unreal Engine 4 by declaring which graphical environment to generate. The Unreal Engine 4 physics system is also utilized for AirSim object attributes and interactions. AirSim also uses OpenCV for object detection/ image recognition in the Unreal Engine 4 environment.

3.4 Communications Interfaces

The system does not currently use any communications interfaces.

4. System Features

4.1 Ground Station/ Control

4.1.1 Description and Priority

Ground Station is the user/ computer that is used. Ground Station is what transmits the mission task input and what receives the sensor data and swarm status.

4.1.2 Stimulus/Response Sequences

Action: User issues mission task for aerial swarm.
Response: Aerial swarm begins mission and reports status of aerial swarm.
Action: Aerial swarm transmits sensor data.
Response: Ground station records sensor data.

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

4.1.3 Functional Requirements

[Req 5] Ground station shall have the ability to assign missions to the aerial swarm.
[Req 6] Ground station shall receive reports on the aerial swarm status including status of all individual UAV.
[Req 7] Ground station shall receive sensor data from aerial swarm.
[Req 8] Ground station shall record sensor data.
[Req 9] Ground station shall display aerial swarm status including status of all individual UAV.

4.2 UAS Swarm

4.2.1 Description and Priority

The aerial swarm is a unified, cohesive grouping of multiple individual UAV. There is a designated lead UAV that leads the swarm and organizes the swarm while operating, collects all the individual UAV statuses, and reports the overall swarm status and individual UAV statuses to the ground station.

4.2.2 Stimulus/Response Sequences

Action: User spawns in an aerial swarm.
Response: First UAV spawned is designated as the lead UAV.
Action: User assigns mission task to aerial swarm through script.
Response: Lead UAV organizes and coordinates with other UAVs to complete mission.
Action: Object identified for analysis.
Response: Aerial swarm maneuvers to object. Aerial swarm measures physical dimensions of object and calculates the volume.

4.2.3 Functional Requirements

[Req 10] The aerial swarm shall designate a lead UAV for swarm organization and communication.
[Req 11] The aerial swarm shall have at least three UAV for any given mission.
[Req 12] The lead UAV shall receive status data from all individual UAV every 0.1 seconds.
[Req 13] The lead UAV shall transmit status data of the aerial swarm and all individual UAV.
[Req 14] The aerial swarm shall measure the volume of an identified object.
[Req 15] The aerial swarm shall transmit sensor data to a repository in the ground station.
[Req 16] The aerial swarm shall determine the positioning of individual UAV and transmit the data to individual UAV.
[Req 17] The aerial swarm shall adjust and continue the mission task when any individual UAV becomes inactive.
[Req 18] The aerial swarm shall return to “home” location when mission task is complete.

4.3 Individual UAV

4.3.1 Description and Priority

An atomic vehicle of the aerial swarm. Individual UAV will have a position and status that is communicated with the swarm. Each UAV will also have some payload that includes sensors such as a camera.

4.3.2 Stimulus/Response Sequences

Action: Lead UAV receives mission task.
Response: Individual UAV follow lead UAV along path.
Action: Swarm formation changed based on task.
Response: Lead UAV sends position coordinates to individual UAV for specified formation.
Action: Status of UAV requested.
Response: UAV sends status to lead UAV.

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

2nd Semester:
Action: System requests sensor data.
Response: Lidar and camera sensor return new data.
Response: System compares sensor data sets.
Action: Data from either sensor is invalid.
Response: System only utilizes data from valid sensor.
Action: UAV receives new movement vector.
Response: UAV orients to face direction of the movement vector.

4.3.3 Functional Requirements

[Req 19] Individual UAV shall communicate position with the aerial swarm using North, East, and Down coordinates.
[Req 20] Individual UAV shall communicate active status with the aerial swarm.
[Req 21] Individual UAV shall avoid collisions with objects including other UAV.
[Req 22] Individual UAV shall have a minimum of a lidar sensor.
[Req 23] Sensor data shall be routed through the aerial swarm via the lead UAV.

2nd Semester:
[Req 24] Individual UAV shall carry a payload that will house sensors.
[Req 25] Individual UAV shall have a camera sensor.
[Req 26] The camera sensor shall validate the lidar sensor data.
[Req 27] When the lidar sensor data is invalid, the camera sensor shall be used exclusively.
[Req 28] When the camera sensor data is invalid, the lidar sensor shall be used exclusively.
[Req 29] Each UAV shall rotate to face the direction of travel (orientation).
[Req 30] UAV re-orientation shall occur for each new path vector. Refer to Figure 4 for a visualization.

4.4 Collision Avoidance (2nd Semester)

4.4.1 Description and Priority

High priority. The aerial swarm, or single UAV, must be able to travel within an environment without colliding with any obstacle. In addition, the avoidance maneuver must follow FAA right-of-way rules [5].

4.4.2 Stimulus/Response Sequences

Action: Collision Detection module receives sensor data.
Response: Collision Detection compares data with avoidance distance.
Action: Collision Detection module identifies potential collision.
Response: Avoidance function activated.
Response: Data comparison operation terminated.
Action: Avoidance maneuver successful.
Response: UAV/ Aerial Swarm continues course to previous waypoint.
Action: Obstacle is detected in front of UAV.
Response: UAV uses right-hand avoidance maneuver.
Action: Obstacle is detected to right of UAV with a leftward velocity.
Response: UAV uses right-hand avoidance maneuver.
Action: Obstacle is detected to left of UAV.
Response: UAV does not use right-hand maneuver.

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

4.4.3 Functional Requirements

[Req 31] A UAV shall avoid collisions with all other objects while moving.
[Req 32] A UAV shall receive data about obstacles in the environment.
[Req 33] The aerial swarm shall avoid collisions with all obstacles while operating.
[Req 34] The aerial swarm shall receive data about obstacles in the environment.
[Req 35] The aerial swarm shall not consider composite UAV as obstacles for other composite UAV.
[Req 36] The Collision Avoidance system shall continuously execute.
[Req 37] The Collision Avoidance system shall identify an impending collision.
[Req 38] The Collision Avoidance system shall activate the avoidance algorithm when an impending collision is detected.
[Req 39] The Collision Avoidance system shall not activate the avoidance algorithm if an impending collision is not detected.
[Req 40] The Collision Avoidance system shall end the comparison operation with sensor data after the avoidance is activated.
[Req 41] The UAV or aerial swarm shall continue the original mission path after successful collision avoidance.
[Req 42] The Collision Avoidance system shall record each impending collision.
[Req 43] The Collision Avoidance system shall record every execution of the avoidance algorithm.
[Req 44] The Collision Avoidance system shall record any collision between the UAV and another object.
[Req 45] The avoidance algorithm shall choose an avoidance path without immediate obstacles.
[Req 46] The avoidance algorithm shall choose an avoidance path that results in a right-hand turn.
[Req 47] The Collision Avoidance system shall manuever according to the Head-On right of way rule shown in Figure 1.
[Req 48] The Collision Avoidance system shall manuever according to the Overtaking right of way rule shown in Figure 2.
[Req 49] The Collision Avoidance system shall manuever according to the Converging right of way rule shown in Figure 3.

5. Other Requirements

5.1 Performance Requirements

[Req 50] The aerial swarm shall continue task and attempt to finish the mission queue when any individual UAV becomes inactive.
[Req 51] The aerial swarm shall report to ground control when a UAV becomes inactive.

5.2 Safety Requirements

We do not have any safety requirements.

5.3 Security Requirements

We do not have any security requirements.

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

5.4 Software Quality Attributes

Adaptability: Modularity facilitates high cohesion and low coupling attributes for our system.
Availability: Currently our system is located in the private Github. All foundation software is open source.
Flexibility: Aerial swarm may be able to continue mission with multiple UAV for given tasks. The aerial swarm must successfully avoid collisions with other objects.
Maintainability: Code commenting, and naming conventions allow for readability. System modules are modular classes/ functions.
Reusability: Each modular class or file may be reused separately.
Testability: Unreal Engine 4.27, Microsoft's AirSim 1.6.0, Visual Studio 2019, and additional tools are necessary for executing our system. Modular classes and functions allow for testing small portions of the system individually.

5.5 Business Rules

There are no business rules for this system.

Appendix A: Glossary

Active:	UAV is operational and in flight.
Inactive:	UAV is not operational or is not in flight.
UAV:	Unmanned Aerial Vehicle.
UAS:	Unmanned Aerial System – "aerial swarm".
VLOS:	Visual Line of Sight
BVLOS:	Beyond Visual Line of Sight

Appendix B: Analysis Models

Analysis Models not included in this document may be found in the System Design Document [4].

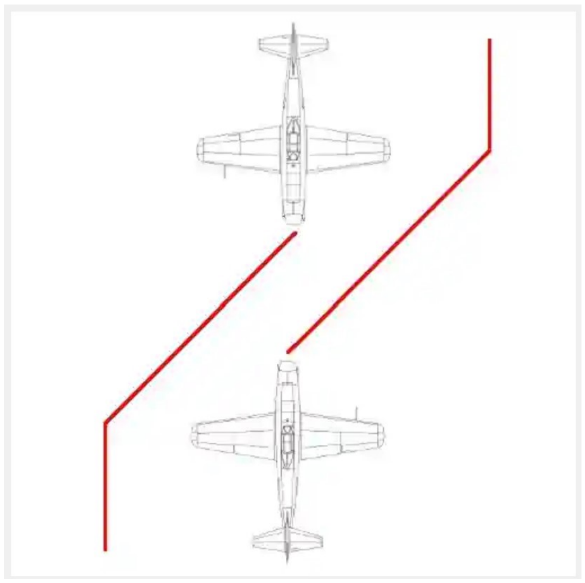

Figure 1. Head-On collision scenario [5]. Requirement 47. No aircraft has right-of-way. Both aircraft must maneuver to the right.

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

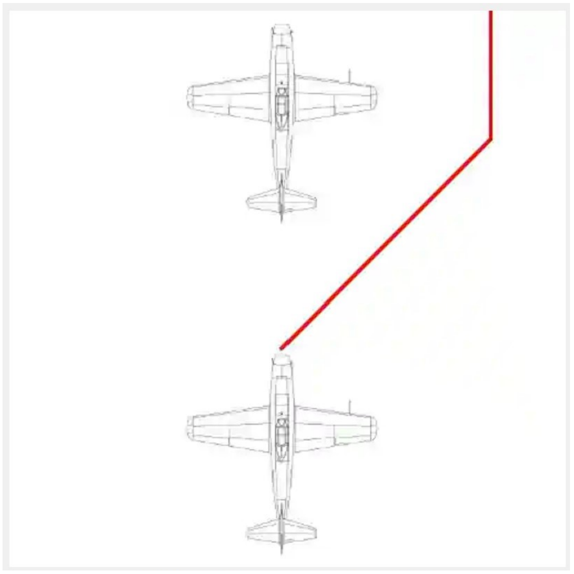

Figure 2. Overtaking scenario [5]. Requirement 48. The slower, front aircraft has right-of-way. The faster, rear aircraft must maneuver right to pass.

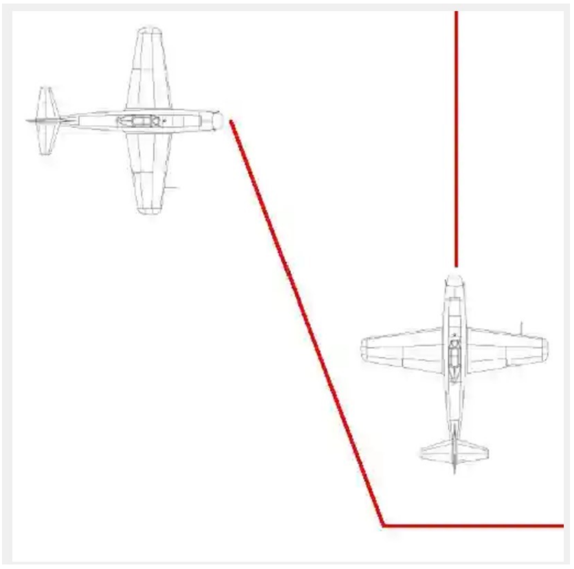

Figure 3. Converging scenario [5]. Requirement 49. The aircraft to the right has right-of-way. The aircraft on the left must maneuver to the right.

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

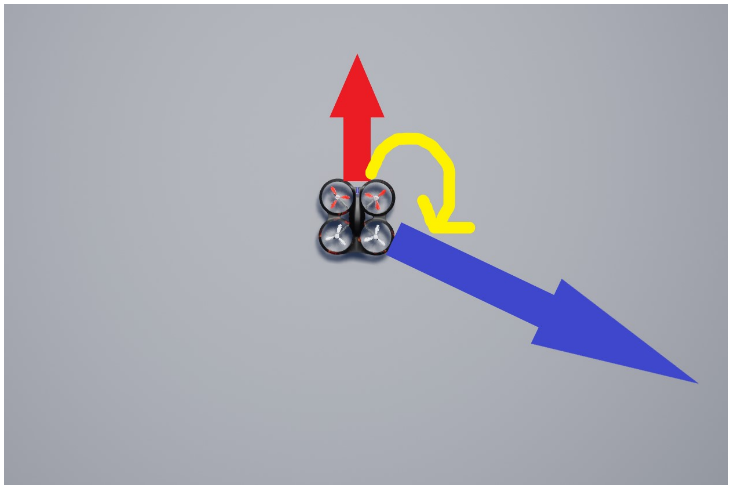

Figure 4. UAV yaw orientation. Requirement 30. In AirSim a UAV moves in any horizontal direction without changing orientation. This figure is an example where a UAV has an original heading of the red vector and a direction of movement as the blue vector. To properly use the forward sensors, the UAV orients the red vector with the blue vector.

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.