Project Report
TIP-166

Advanced Air Mobility Assessment Framework: FY21 HP/ATC/HADR Technical Investment Program

L.E. Alvarez
T.A. Bonin
J.C. Jones Jr.

17 March 2022

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

This material is based upon work supported by the United States Air Force under Air Force Contract No. FA8702-15-D-0001.

This report is the result of studies performed at Lincoln Laboratory, a federally funded research and development center operated by Massachusetts Institute of Technology. This material is based upon work supported by the United States Air Force under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force.

Massachusetts Institute of Technology
Lincoln Laboratory

Advanced Air Mobility Assessment Framework: FY21 HP/ATC/HADR Technical Investment Program

L.E. Alvarez
Group 42

T.A. Bonin
J.C. Jones Jr.
Group 43

Project Report TIP-166

17 March 2022

Lexington Massachusetts

This page intentionally left blank.

TABLE OF CONTENTS

		Page
	Abstract	v
1.	INTRODUCTION	1
2.	MODELING	3
	2.1 General Upgrades	3
	2.2 Network Design	4
	2.3 Discrete Event Simulation	11
	2.4 Weather Simulation and Impacts	12
3.	FUTURE WORK	25
A	BLUE SKY INTEGRATION	27
	References	33

This page intentionally left blank.

ABSTRACT

Advanced Air Mobility encompasses emerging aviation technologies that transport people and cargo between local, regional, or urban locations that are currently underserved by aviation and other transportation modalities. The disruptive nature of these technologies has pushed industry, academia, and governments to devote significant investments to understand their impact on airspace risk, operational procedures, and passengers. As part of FY20 funding of this program, a flexible framework was designed to assess the operational viability of these technologies and the sensitivity to a variety of assumptions [1]. During the second year of funding (FY21), this framework was used to simulate an initial AAM implementation scenario in New York City. This scenario was created by replacing a portion of NYC taxi requests with electric vertical takeoff and landing vehicles. The framework was used to assess the sensitivity of this scenario to a variety of system assumption. The main focus of FY21 was to integrate weather impacts, such as shutdowns of routes or tail or head wind impacts during flight to calculate new routes.

This page intentionally left blank.

1. INTRODUCTION

Advanced Air Mobility (AAM) is a technology that industry and government agencies are actively pursuing to transport people, goods, and services, with aircraft ranging in size from handheld unmanned air vehicles to large passenger-carrying vehicles. Within AAM, the subset of operations known as Urban Air Mobility (UAM) are focused on operations above urban environments. Air travel within major urban environments is not a new concept: in the 1970s, helicopter commuting was a common means of transportation to and from John F. Kennedy international airport, however safety concerns restricted operations from flying over the city [2]. Recent advancements in technology has reawakened interest in this urban mode of transportation to operate within cities [3]. Market research shows a potential for millions of operations of unmanned and manned UAM aircraft at low altitude on an annual basis. This anticipated amount of on-demand air traffic is expected to stress the U.S. National Airspace System in an unprecedented manner [4].

To safely integrate this large number of operations, government entities have supported testing and evaluation and provided an initial stand on operating procedures of UAM technologies through efforts like the NASA AAM National Campaign [5] and the FAA Urban Air Mobility Concept of Operations [6]. Through the AAM National Campaign, NASA is predominantly focused on information sharing between industry partners and government entities to understand the gaps that impeded operations beyond UML1 as described in [7]. As part of this program, four working groups have conducted public discussions with industry, academia, and national laboratories. These working groups focus on aircraft certification, airspace, community integration, and crosscutting impacts (e.g., noise abatement via aircraft design to have community acceptance). The FAA has partnered with several industry members as well as university-led initiatives to research the regulatory gaps and understand the consequences of changes to regulations. In alignment with these initiatives, several industry members are implementing the UAM vision by engaging in trial operations, releasing operational concepts, and studying vehicle design requirements [8], [9], [10], [11], and [12]. The vast majority of these concepts have focused on transporting three to five people, in vehicles with operating ranges less than 100 miles. Although the majority of concepts envision the use of electric vertical takeoff and landing (eVTOL) aircraft, current operations make use of helicopters for testing procedures and market feasibility [13]. At the time of writing this report, congress is in the process of reviewing a bill for an interagacengy working group, led by the department of transportation, to established a cohesive national strategy for AAM [14].

Within standard development organization subcommittees, such as RTCA SC-228, industry partners (e.g., Joby, and Wisk) have proposed initial concepts of operations as part of UML1 operations or more complex operations in their AAM concept of operations. The FAA CONOPS [7] and RTCA SC-228 envision AAM traffic flow to be contained within specified corridors, where Air Traffic Control (ATC) would not provide active support for these vehicles. The initial operations of UAM aircraft along these corridors would be akin to visual flight rule (VFR) operations [15]. Operations within these corridors would be expected to have strategic conflict management either through a centralized management system such as a UAS Traffic Management (UTM) system, or decentralized operator based system [16]. NASA, on the other hand, is focused on a future UML4 CONOPS where the UAM Operating Environment (UOE) would allow flights to be strategically

deconflicted with early versions being akin to corridors, but evolving to allow operations anywhere in the UOE if properly separated. In addition, unlike commercial air travel, the landing and takeoff locations (vertiports) of UAM aircraft will likely be built through private capital [17]. Challenges facing the design of these operations range from environmental impacts to the underlying population (e.g., noise impacts), weather shutdowns, emergency landing site availability, operating flight rules, airspace capacity, and landing and takeoff locations [18]. The throughput of these concepts of operations is being researched by NUAIR in collaboration with NASA to develop landing/takeoff procedures and explore the differences associated with heliport concepts [19]. Ultimately, to design robust operations, government and industry must understand complex infrastructure interdependence of corridor network design, landing and takeoff site capacity (vertipads), schedule assurance, and maximum sustainable throughput of the system.

In year one, the UAMToolKit was developed to support analysis of infrastructure requirements for AAM airspace integration. The reader is directed to the first year report which describes the UAMToolkit framework for assessing large scale, realistic operations of UAM and the submodules created to for demand, scheduling, and event modeling [1]. In this report, we describe the integration of weather impacts on UAM operations and the modeling design decisions that were made to enable a fast time simulation environment. This report also summarizes the network structure and its accompanying fleet-to-vertiport ratios required to accommodate an average passenger delay below 15 minutes; additional delay time would be expected to impact passenger willingness to travel [20], [21]. A full analysis utilizing the UAMToolKit of the New York City airspace can be found in [22].

2. MODELING

The simulation framework is composed of a demand model, a scheduler, weather module, re-scheduler, and a discrete-event simulator. A demand model composed of Poisson distributions of demand as a function of location drives the traffic requests for UAM. This differs from previous studies that used a synthetic probability distribution to drive events [23]. Initial research in FY20 used a greedy scheduler to assign aircraft to the demand based on the state of the closest available aircraft; in FY21, initial work with MIT Sloan used more advanced optimization of the schedule to be robust to weather and reduce conflict management intervention along the route. The weather simulator as explained in Section 2.4 allows for better calculation of the arrival time of aircraft and whether or not the aircraft is allowed to operate given some aircraft thresholds. A discrete event simulator then ingests the scheduled traffic and creates a sequence of events that are reassessed at each time step.

2.1 GENERAL UPGRADES

In year two of the program, we focused on making the Python package stand alone such that someone can download the package and call modules independently. During the large scale analysis conducted as part of an AIAA Aviation Forum publication [22], a few improvements were made to the demand model as described in the next section.

2.1.1 Demand Modeling

During our large-scale analysis, it was found the demand model was producing more requests in regions of the city where historical data showed request values lower than one passenger request per minute. This was caused by the use of the Poisson distribution which requires an integer value for the frequency of events every minute. Because of the need to have an integer value on the frequency parameter of the Poisson distribution, it could be rounded up or down to the next integer values. In the extreme scenarios, a frequency of 0.51 requests per minute would be interpreted as a Poisson distribution with a integer value of 1 and when querying the number of requests per hour, it would most likely lead to 60 passengers per hour as opposed to the historical average of 30 passenger request per hour. It also presents an issue because values smaller than 0.5 have a mean of zero request. This in turn is not representative as it leads to a right-tailed distribution of the actual demand we are trying to model.

For this reason, a second demand querying approach was adopted in which the demand frequency parameters are combined into the nearest vertiport pair for a larger frequency count per minute per Origin-Destination (O-D) pairs. That is, instead of looking at each individual cell as depicted in Figure 1 where it is queried for demand from origin grid cell to destination grid cell, the cells are assigned to a vertiport. This allows us to have larger integer values and avoid the close to zero issue previously presented. Since our data will continue to have fractional values, the integer value used in a Poisson distribution is separated into the actual integer value (f) with a modifier (U) as represented in equation 1.

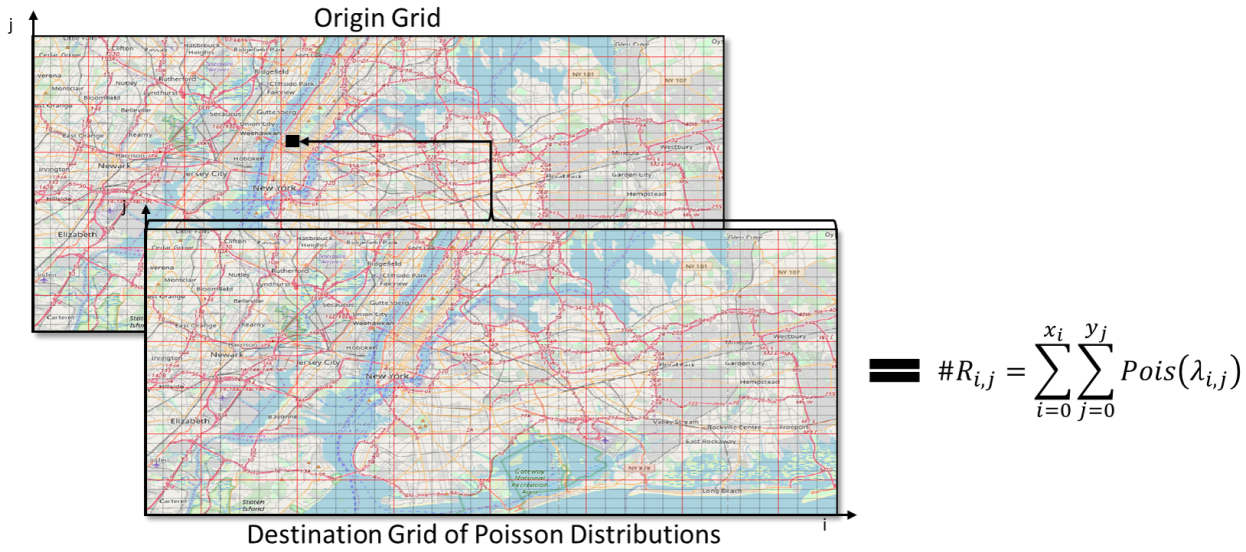

Figure 1. Mapping of origin-destination pairs by representing requested destination as a grid world with a set of Poisson distributions and attributing the cumulative requests to the origin grid world.

$$\#Passengers = Poisson(\lambda) \mid \lambda = f + U(0,1) \tag{1}$$

2.2 NETWORK DESIGN

To calculate distance between vertiports, either direct flight paths between vertiports (Figure 2) or a system of nodes and edges that represent flight paths (Figure 3) is used. Building on the modeling capability of year one, the network module was extended with utility functions that are common for manipulating the networks designed by previous users. Previously, the users would have to recreate a network entirely if a new set of vertiports was desired for testing. The network utility class is now able to save a vertiport agnostic network on file for any user to use and merge with vertiport sites that the user desires. An example is provided below of how a user can connect an independent network with a set of vertiport locations. The second example shows how a direct vertiport-to-vertiport network can be easily created with the network utility function.

```
import networkx as nx
import numpy as np
import os
import json
import csv
import matplotlib.pyplot as plt
from math import radians, degrees, sin, cos, asin, acos, sqrt
from networkUtilities import *
```

```
networkType = 1
image = 'nyc_im.png'
BBox = 'nyc_bbox.npy'
BBox = np.load(BBox)
image = plt.imread(image)
numVaptOrg = 29
numVaptNew = 29
vpt_loc = np.load('vertiport_29.py')

# what network type is this 1) helo netowrk, 2) direct network
if (networkType == 1):

    # original network only has 15 vertiports attached to it
    originalNetwork = 'checkpoint1.gml'

    # future node connections we would like to modify network with
    pairs = [(57,69),(85,73),(87,61),(100,83),(75,83),(161,168)]

    # save name for the network file
    saveLocation = '.'

elif (networkType == 2):
    originalMap = '' # not required since starting from scratch
    pairs = [] # be used to connect all vertiports
    saveLocation = '.'

if (networkType==1):

    # load the old network
    H = nx.read_gml(originalMap)

    # list of nodes associated with vertiport
    removeList = list(range(numVaptOrg))

    # new vertiport node list
    vapt_list = list(range(numVaptNew))

    # create Independent Network
    createIndependentNetwork(H,pairs,removeList,saveName, BBox, image)

    # add the vertiports you want (for now we can only add up to 29 vertiports otherwise I have to shift
        ↪ the nodes)
    addVertiportsToNetwork(vpt_loc,H)

    # connect to nearest node the nearest
    connectNearestNodeToVertiport(H, vapt_list, vpt_loc)

    # custom connections I desire for this network
    addEdges(pairs,H)
```

```
elif (networkType==2):

    # load empty network
    H = nx.Graph()

    # add vertiports to empty network
    addVertiportsToNetwork(vpt_loc,H)

    # create a list of pairs of all vertiports
    vapt_list = list(range(numVaptNew))
    for i in range(numVaptNew):
        for j in range(numVaptNew):
            if(i!=j):
                pairs.append((i,j))

    # connect all vertiports
    addEdges(pairs,H)

# plot the network
simplePlot(H, BBox, image, vapt_list)

# save all files required by UAMToolkit
createSchedulerFiles(H, saveLocation)
```

The network module was also given the capability of assigning properties to each edge connecting node of the network. With this capability, we can assign a number of aircraft that are allowed to traverse the edge, envisioned for use in separation assurance applications. Edge properties are currently used as part of the network to assign a Boolean denoting weather conditions between those nodes exceeds the weather threshold the user requires during flights. This also allows the weather module to calculate the weather impacts along a route by assigning properties to the network—for example assigning wind magnitude and direction at each node of the network or assigning the average wind condition along the edge connecting the nodes.

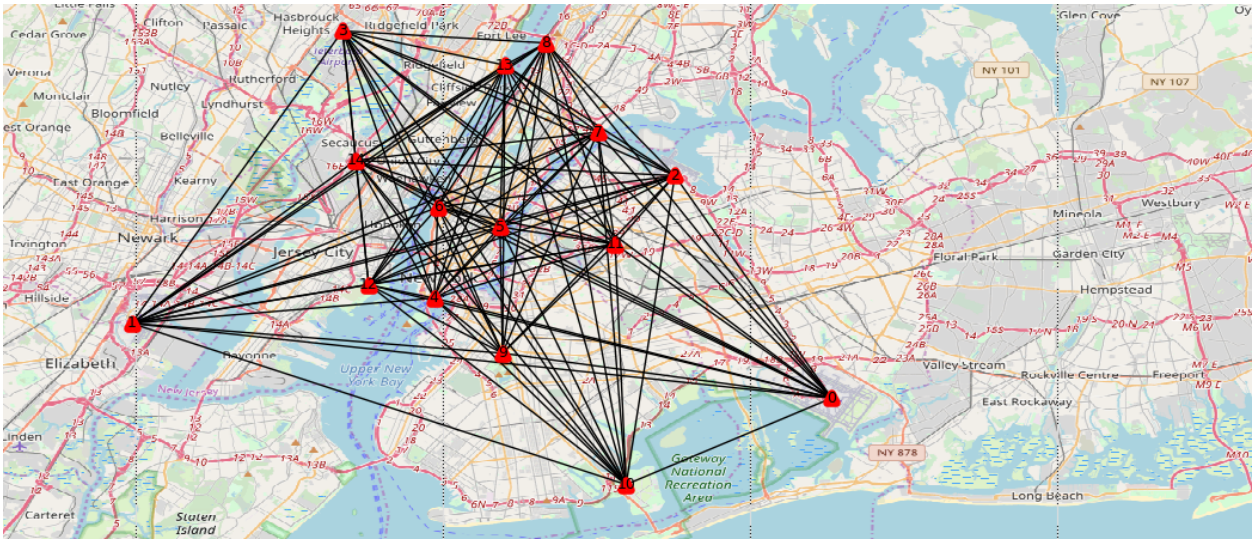

Figure 2. Direct travel corridor network.

Figure 3. Helicopter route network constructed for analysis of UAM operations using existing airspace infrastructure.

2.2.1 Route IDs

As part of the continued development in FY22, the capability was added of creating a set of route IDs that can be interchanged between partners and MIT LL to specifically simulate the routes that external partners desired given a network. Previously, the discrete model and schedulers would assume the shortest route in a network would always be utilized to calculate the total flight time. In particular, the route ID is important when a study considers the weather impacts to the aircraft, whether it be during flight or as part of the gate holding that is handled by the vertiport model in the discrete event module. To accomplish this task, a user would simply call the *all_simple_paths* from a source to a destination and can then downsample to the N shortest routes, if desired. A sample of how this has been done in the past can be found here:

```
import networks.sectorsUtils
import networkx as nx
import networks.networkUtilities as nxu

# helper files
spFile = "shortestPath.json"
plFile = "shortestPath_len.json"
netFile = "vfrLatest.gml"
vertiport_path = "vertiport_29.npy"

# load information needed
net = nx.read_gml(netFile) # networkx object

# list of vertiport index
vptList = []
for i in range(29):
    vptList.append(str(i))

routeIdSet = {}
routeIdLength = {}
routeIdPath = {}
firstSet = True
id = 1
for orig in vptList:

    for dest in vptList:
        if dest == orig:
            continue
        for path in nx.all_simple_paths(net, source=orig, target=dest):
            # removing origin and destination to reduce paths
            setPath = set(path[1:-1])
            # remove flying through vertiport nodes
            noVertiport = not(any(setPath.intersection(vptList)))
            if (firstSet & noVertiport):
                routeIdSet[id] = setPath
                routeIdPath[id] = path[1:-1]
                routeIdLength[id] = nxu.getTruePathLength(net, path)
```

```
        id += 1
    elif (noVertiport):
        foundSet = False
        for k in routeIdSet.keys():
            foundSet = routeIdSet[k] == setPath
        if not(foundSet):
            routeIdSet[id] = setPath
            routeIdPath[id] = path # to remove od from path[1:-1]
            routeIdLength[id] = nxu.getTruePathLength(net, path)
            id += 1
firstSet = False

# JSON file saves
with open("routeIdSet.json","w") as outfile:
    json.dump(routeIdSet, outfile)
```

2.2.2 Sector conversion

During the integration of the weather module, it was evident that the impact of weather on the operations might be dependent on the resolution of the network that is under evaluation. For example, the network shown in Figure 4 has tightly spaced nodes to more precisely show the exact path that should be taken by the aircraft. This network would provide weather data at higher resolution than the direct route network shown in Figure 2. Thus, it was decided to average the weather conditions by discritizing the networks with nodes spaced by 100 m and representing the data in a sector graph where the edge has an attribute of the weather experienced between the nodes it connects. The sector graph only contains a single edge that is not shared between paths, thus reducing the number of data points that need to be maintained and providing a quicker way to query if the weather is above a specified user threshold along the flight path. The resulting sector graph would similarly show the overall structure of the network; however it is evident that some of the data can be lost, as seen by the relatively sparse number of nodes in Figure 5.

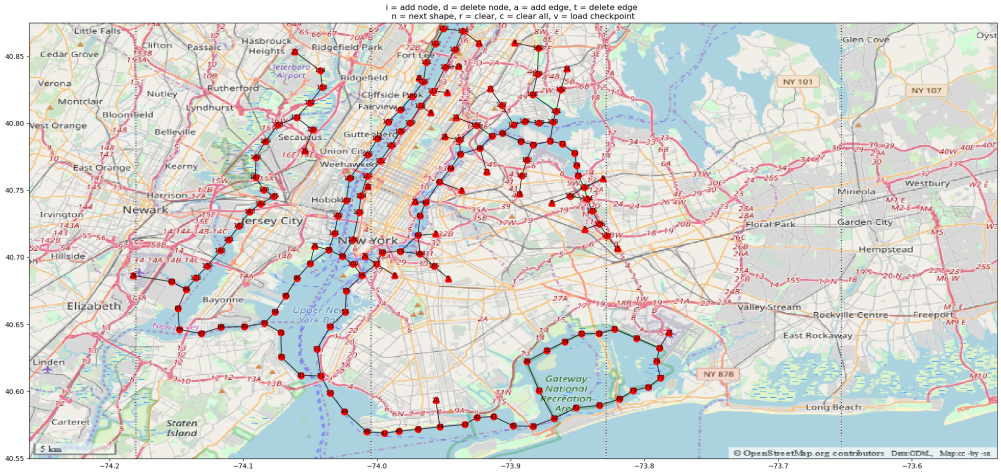

Figure 4. Example of network with tightly spaced nodes describing the structure of the network.

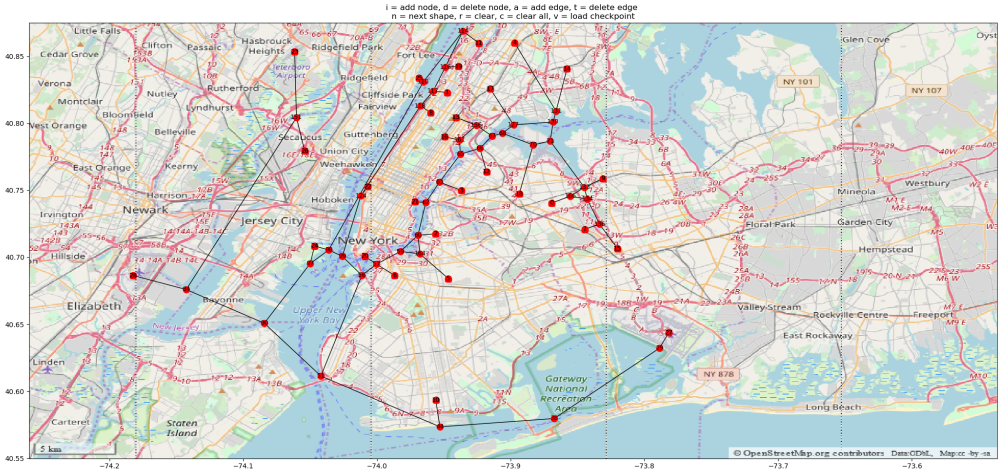

Figure 5. Example of the sector representation of the previous network.

2.3 DISCRETE EVENT SIMULATION

A discrete-event simulation model was developed to analyze the system performance of UAM traffic networks. Applying a discrete-event model allows the system to jump to the next time step where a new event occurs, with the assumption that the state of system does not change outside of the event queue. This simulation supports analysis of various scenarios by modeling the traffic events defined in the schedule. A scenario is defined by configuring the number of vertiports, the number of vertipads at each vertiport, and the vertiport locations. The user can define the vehicle characteristics by specifying the number of aircraft in the fleet, the speed of each aircraft, the number of seats on each aircraft.

The simulation also incorporates constraints such as the airborne holding time at the vertiport prior to diverting and the minimum amount of time each aircraft needs to spend on the ground before departure. At the beginning of each simulation, aircraft are placed at their initial vertiport defined in the schedule. Flights are scheduled on-demand and travel between vertiports in order to meet the requested trip demand using the scheduler described in [1] or the max passenger throughput optimization as described in Section 2. At each vertiport, passengers are loaded onto the vehicle prior to departure. The time associated with loading can be represented as a constant or by a uniform distribution. When a flight departs from a vertiport, it flies along the set of waypoints within the flight plan. At each waypoint, an event is triggered and a new decision is made regarding where to proceed. If the aircraft receives no new information, it will proceed along its current flight plan. If an amendment to the flight plan is received, the flight will alter its destination based on the information in the amended flight plan. When a flight approaches the destination vertiport, it will land if there is a vertipad available and passengers are dropped off, otherwise it will hold until one becomes available. A diagram of the simulation framework is shown in Figure 6.

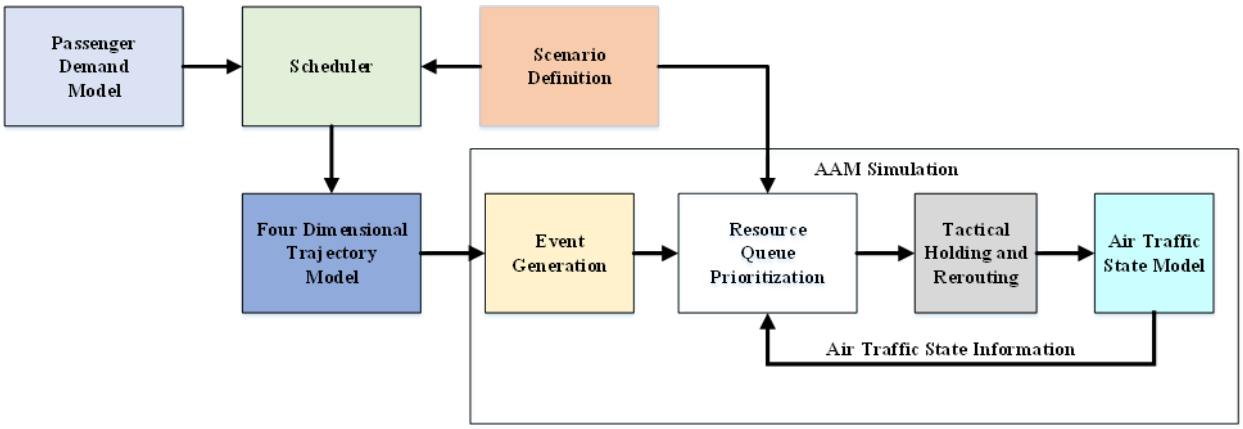

Figure 6. The Advanced Air Mobility Simulation Framework.

The simulation evaluates the decisions of the users through a set of operational metrics including passenger delay, ground delay, air delay, vertiport utilization, and flight diversions. At the

end of the simulation, the users can view the performance of the scenario relative to these metrics. The total simulation computation time of the system is on the order of 20 minutes with the discrete model encompassing 2 minutes of simulation. By iteratively running different configurations against various scenarios, users can gauge the effectiveness of potential interventions and gain insight into how to tailor the schedule, fleet size, and vertiport structures to optimize the metrics of greatest importance for a given passenger demand distribution. They can also perform system trade studies to understand the relative variation between scenario parameters. They can use the results of these studies to configure the AAM network to meet specific performance requirements.

2.4 WEATHER SIMULATION AND IMPACTS

In order to incorporate weather impacts into the UAM simulation, the general approach shown in Figure 7 was taken. High-resolution weather data files were created for retrospective case days using the open-sourced Weather Research and Forecasting (WRF) model [24]. The wrf-python package was used to extract weather information at nodes and vertiports at different altitude bins, with this data saved as JSON files containing the simulated weather for the entire 24-hr simulation period. This weather information was used to update the travel time incorporating wind impacts, as well as assess which trips could be completed based on weather conditions at the time of the scheduled flight. These results were then synthesized to create impact metrics.

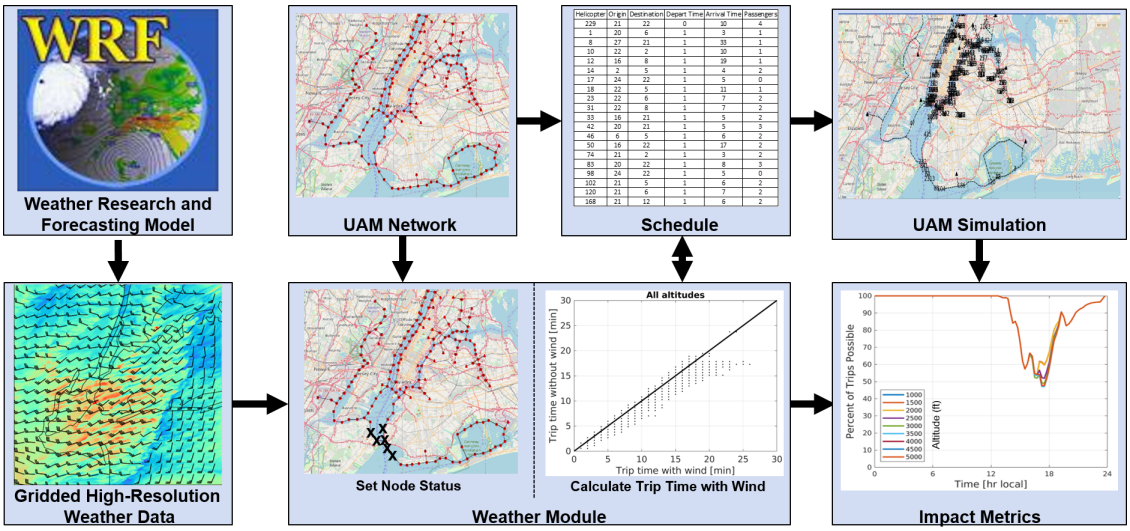

Figure 7. Flow diagram showing the approach for incorporating weather impacts into the UAM Simulator.

2.4.1 Weather Simulation Setup

The open-sourced WRF codebase was leveraged for this effort to provide the high-resolution weather information incorporated into the UAM Simulation. The WRF model is primarily maintained and supported by the National Center for Atmospheric Reserearch (NCAR), with contribu-

tions from various users and developers around the globe. The software package is regularly updated and released with new features, parameterizations, and optimized computations after rigorous testing by the development team. WRF also serves as the framework for the High-Resolution Rapid Refresh (HRRR) model operationally run by the National Oceanic and Atmospheric Administration (NOAA), which is widely used for a variety of applications and considered the state-of-the-art operational model. For these reasons and common use in the meteorology community, WRF was chosen for simulating the weather conditions around a domain encompassing the UAM network.

Running WRF is highly customizable and can be easily modified. The resolution of the model, domain, physical parameterizations, forecast window, and interval are all chosen and set prior to running it. WRF simulates the weather conditions based on initial and boundary conditions and produces weather data—including temperature, wind, humidity, precipitation, and many other diagnostic fields—on a full 3-D volumetric grid. Meteorological observations can be optionally assimilated into the model through the supplemental WRF Data Assimilation package. A more in-depth overview of WRF in general can be found in [24], and a description of how WRF has been implemented at MIT Lincoln Laboratory on LLSC is given in [25]. For purposes here, the primary limitations of WRF are that it does not have any representation of individual buildings and cannot simulate flow around them, and it cannot model very-fine-scale turbulence.

For the WRF retrospective forecasts produced and used here, three nested domains were set up centered on NYC, as shown in Figure 8. The domains D01, D02, and D03 respectively had resolutions of 2.5 km, 500 m, and 100 m. By using a nested configuration, each domain was able to initialize using coarser boundary conditions, "spin-up", and resolve increasingly fine-scaled processes to more accurately represent down-scaled features for a larger region of the domain by limiting edge effects at the boundary of each. This enables a more realistic simulation of atmospheric processes in the fine-scale D03 domain. The extent of D03 encompassed all of the networks described in Section 2.2, plus a buffer to account for future networks that might be larger and to account for spin-up of features flowing into the domain. The WRF output on the 100-m domain was used as the weather data input into the weather module of the UAM simulation. As mentioned earlier, WRF cannot resolved fine-scale turbulent features but it does explicitly resolve large turbulent eddies in the 100-m domain.

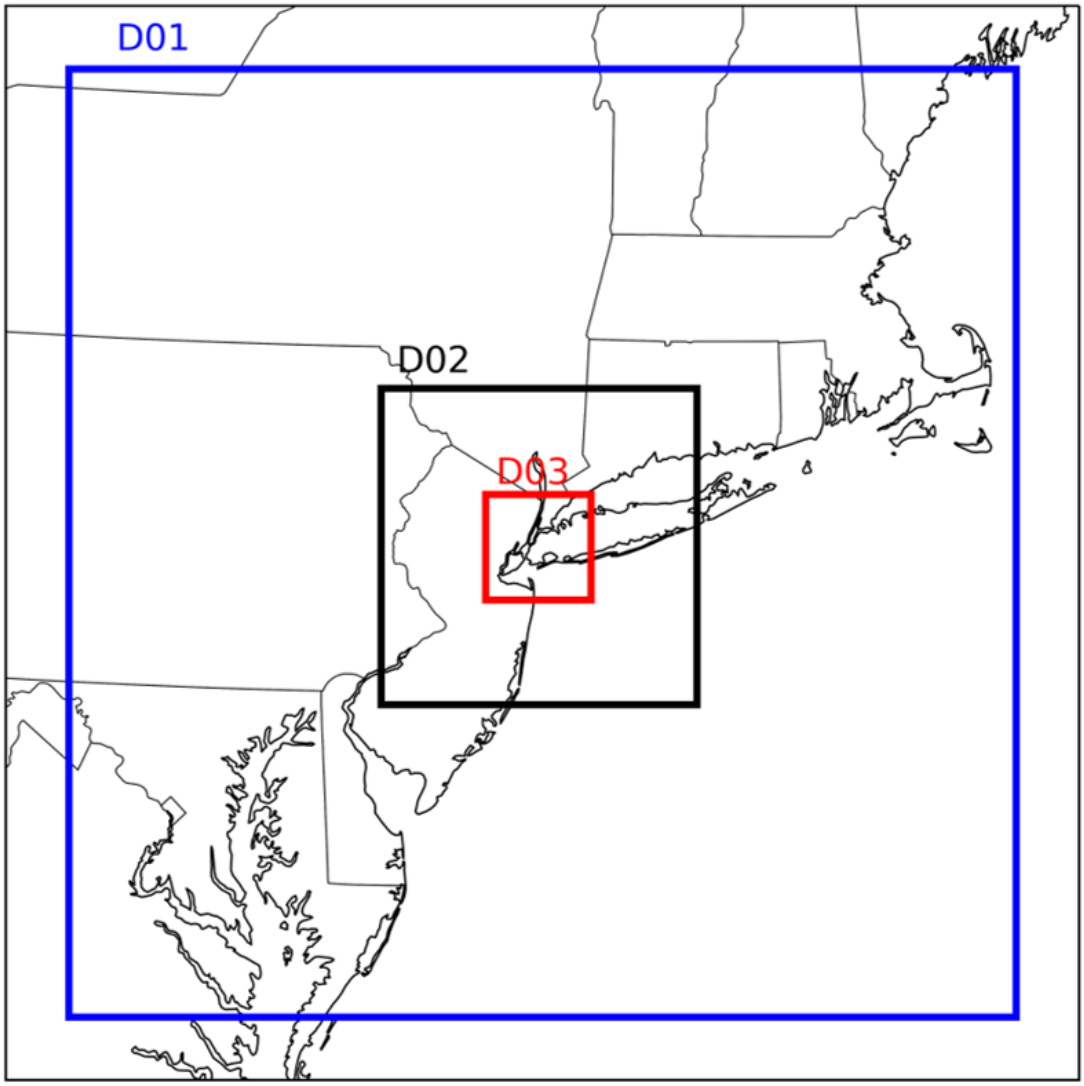

Figure 8. Map showing the extent of the nested WRF domains centered on NYC.

In the WRF runs, data on D03 was output at 15 min forecast intervals to capture rapidly varying conditions. Outputting at a higher frequency has diminishing returns due to the significant amount of disk space and longer processing time in the UAM simulation weather module, especially given that there is more correlation between forecasts as the interval becomes shorter. Model analyses of the weather from the 3-km HRRR were used as the initial conditions and as the boundary conditions, updating each hour the model was run to capture weather incoming at the boundaries of the domain. No additional data assimilation of observations was performed—given that a large number of observations were already assimilated into the HRRR, limited improvements would be expected.

2.4.2 Simulation of Weather Impacts

Within the weather module of the UAM simulator, two independent processes were developed to run separately to assess certain weather impacts on UAM flights. One process is to open/close individual nodes and routes based on the presence of hazardous weather conditions, while the other process modifies the flight schedule and updates the time for a trip to be completed to include wind effects on the aircraft along the 4-D trajectory. They were developed to be independent so that each can be evaluated depending on the objective of the analysis performed, although the sector status information can be optionally ingested into the trip time calculation. A diagram showing this framework of the weather module is provided in Figure 9.

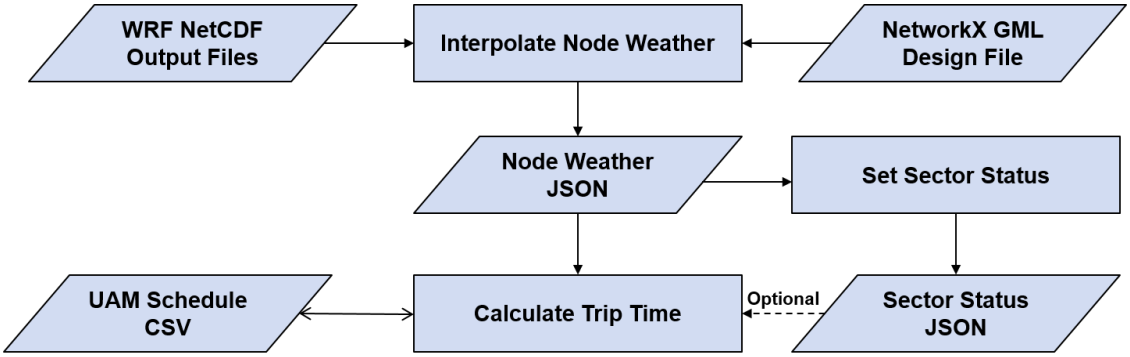

Figure 9. Flow diagram showing the framework of the UAM simulation weather module.

As shown in this diagram, the primary data used by the weather module include a NetworkX Graph Modelling Language (GML) file containing the network graph as well as a set of WRF output Network Common Data Form (NetCDF) files at each forecast horizon. The first process loads in the nodes and vertiports including their latitude and longitude files and iterates over a list of WRF NetCDF files over the simulated data to interpolate the weather data in three-dimensions at desired altitudes over each node. Since the WRF files contain a large amount of diagnostics over a full 3-D grid, this process pulls out the pertinent weather data from all the files and consolidates the data into a single output JSON file containing information needed by the downstream processes. Currently, only the horizontal wind and simulated radar reflectivity are retrieved from the NetCDF files and output in the JSON. In the future, additional diagnostics of interest could be added to this process. Additionally, the weather information is output at altitudes spaced every 500 ft, starting at 500 ft above ground level (AGL) upwards to 5000 ft AGL. These altitudes were chosen to evaluate the impact of UAM cooridor altitude on operations, as the CONOPS are still not fully developed and it's unclear at which height these vehicles will fly. Additionally, flights in opposing directions may cruise at different altitudes to ensure vertical separation.

The JSON file containing weather information at the nodes is then used in two separate processes, one of which sets node and sector status depending on the weather conditions. In this process, the capacity factor of individual nodes is set as 1 (fully open) or 0 (fully closed) for each

forecast interval, nominally 15 min in this analysis. In the future, the capacity factor may be set to fractional values to denote conditions where flights may still occur but at reduced tempo, such as during instrument meteorological conditions (IMC) when flights may still be allowed but additional horizontal separation may be required. The nodes are set as either fully open or closed based on whether thresholds are exceeded. Specifically, a node is closed at a given altitude if the wind speed exceeds 50 kts or the radar reflectivity exceeds 40 dBZ. Both of these thresholds are parameterized and can be easily changed as more information becomes known about UAM CONOPS and vehicle limitations. These values were chosen because the wind speed is the estimated maximum wind in which a UAM can operate and the 40 dBZ threshold is typical in convective precipitation and thunderstorms corresponding to a rainfall rate of 0.45 in/hr, which indicate regions of convective storms that UAM aircraft likely would not fly into. The capacity of a sector, which consists of a number of nodes that are serially connected, is set based on the minimum capacity of any node in the sector. The capacity of each sector is output as a JSON file.

Separately, the estimated trip time is calculated for each scheduled flight by incorporating the weather information. This process reads in a standard UAM schedule file and outputs a new schedule file with updated trip times for a variety of departure times—to account for possibly delays—and at each flight altitude. For each scheduled flight, the shortest open route is found for each flight and the 4-D trajectory is used to calculate the trip time. The trip is initiated and the time for the aircraft to fly between a pair of nodes between the origin and destination vertiport is integrated, wherein the ground speed of the aircraft is modified through addition/subtraction of the tailwind/headwind to the airspeed, respectively. Since the aerodynamic characteristics of the vehicles are not yet known, the crabbing effect is not considered in this computation of the trip time. Further, no additional time is added to account for slower vehicle speeds around takeoff and landing. These features can be added in the future.

2.4.3 Case Days

For this analysis, WRF was run for twelve case days, which are listed in Table 1 with a brief description of the weather conditions. The qualitative impact of the weather conditions on each case day is given in a range from low (few/no cancellations, minor wind impact on flight times) to extreme (many cancellations, significant time period where no UAM flights could occur). These case days were manually chosen to represent a variety of conditions including winter weather, severe thunderstorms, tropical storm, fog, sea breezes, and fair weather. A day from each month of the year was chosen to capture a variety of seasons. While only the effects of wind and precipitation (i.e., convection) were incorporated into the UAM simulation weather module so far, the dataset of case days will allow other impacts (e.g., low ceiling and visibility, wintry precipitation) to be analyzed in the future as desired.

2.4.4 Results of Weather Impacts

For each of the case days, the weather impacts on simulated UAM operations were analyzed to produce quantifiable metrics of both winds and heavy precipitation. The two main metrics that were produced were 1) the impact of weather on trip time and 2) the percentage of scheduled trips that could not be completed due to hazardous weather conditions. For the first metric, trip

TABLE 1

Case Days Simulated in WRF

Date	Brief Weather Description	Impact
21 April 2020	Cold front with line of severe storms in the afternoon	Moderate
28 June 2020	Scattered severe storms in the evening	Moderate
10 July 2020	Tropical Storm Fay passed near NYC in the afternoon	Moderate
12 August 2020	Sea breeze in NYC area that formed a few storms	Moderate
15 September 2020	Fair weather and dry with light northerly wind	Low
1 October 2020	Fair weather with light southwesterly wind	Low
3 November 2020	Fair weather with strong westerly winds	High
16 December 2020	Fair weather during day with Nor'easter approaching in the evening	High
2 January 2021	Light rain in early morning, followed by strong winds but dry conditions	High
25 February 2021	Dry cold front in early morning followed by strong northwesterly winds	High
26 March 2021	Dense fog and high winds through noon followed by a few storms	Extreme
19 May 2021	Fair weather and light winds with sea breeze over NYC	Low

times could be lengthened due to the shortest path between origin and destination being closed due to hazardous weather necessitating a longer route being used. Additionally, the trip time for any given flight could be shortened from an advantageous tailwind along the route or lengthened from headwinds. For the second metric, trips could not be completed at their scheduled departure time if there were hazardous weather conditions along all possible routes between the origin and destination. These metrics were produced for each case day and network pair considering all possible corridor altitudes from 500 ft up to 5000 ft AGL. The same schedule (e.g., vehicle airspeed, origin, destination, planned departure time) was used for each simulation to make the results directly comparable between network and days.

Impact of Weather on Trip Duration

Generally, weather had only a small impact on the simulated travel times of flights that could be completed regardless of the network design used or altitude where flights were conducted. This was true even for strong wind days, such as 2 January 2021 as shown in the two-dimensional histograms in Figure 10 and Figure 11 at 1000 ft and 3500 ft, respectively. As shown by the bright colors, most of the flights were short in duration (less than 7 min). These short flights are less sensitive to wind effects (as evidenced by being clustered on the one-to-one line) as there is little distance for changes in the ground speed to accumulate to meaningful differences in the flight duration, especially given that in this simulation the vehicle is assumed to fly at cruising speed between the origin and destination. These results may differ in the future if realistic changes in the airspeed are introduced into the simulation, such as low airspeed around takeoff and landing.

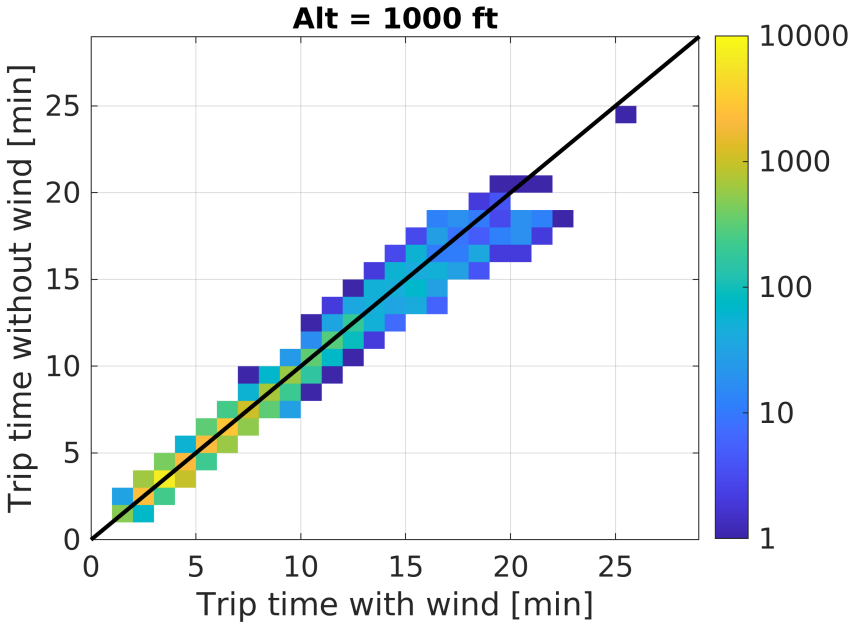

Figure 10. Two-dimensional histogram of the travel time with and without weather effects for scheduled flights on 2 January 2021 on the waterways network at 1000 ft. The color bar indicates the number of flights in each bin in a logarithmic scale. The solid line represents a one-to-one relationship.

By comparing the results in Figure 10 and Figure 11 for the same days at different altitudes, it is clear that there is a stronger impact of winds on travel at higher altitudes in general. This is due to the fact that within the planetary boundary-layer, the wind speed generally increases with height above the surface where friction reduces the wind speed. On this particular day, a local maximum in the wind speed at 3500 ft was present resulting in the greatest impact on travel times compared to other altitudes including up to 5000 ft. This led to a few trips being up to 10 min longer than would be expected if winds were not included (e.g., trips schedules to take 18 min took 28 min with adverse headwinds along the route). While some travel times of flights decreased due to tailwinds (points above the one-to-one line), generally the impact was a slight increase in trip times particularly for the flights that were longer. While these results are only from one day, they are generally applicable across all of the case days examined with varying wind effects dependent on the magnitude of the wind speed at different heights.

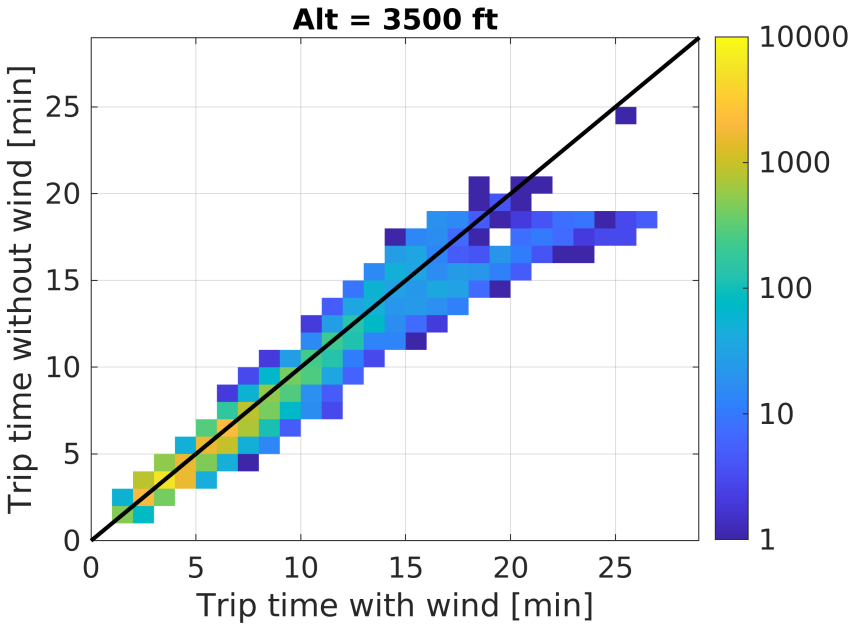

Figure 11. Two-dimensional histogram of the travel time with and without weather effects for scheduled flights on 2 January 2021 on the waterways network at 3500 ft. The color bar indicates the number of flights in each bin in a logarithmic scale. The solid line represents a one-to-one relationship.

Impact of Weather on Ability to Complete Trip

The impact of weather on the ability to successfully complete a scheduled trip was also analyzed for each of the case days. In this section, three of the case days are discussed in detail and a summary of all the days is provided at the end. A scheduled trip was deemed possible if there was an open route between the origin and destination, with routes closed down for strong winds and heavy precipitation at any node along the path as discussed in Section 2.4.2. For each of the cases,

only the results for the waterways network was shown. The general trends were also consistent for the other networks, with some differences in the actual impact on the UAM simulated traffic that is discussed in the summary of cases.

The first case day analyzed is 21 April 2020, as shown in Figure 12. On this day, a strong cold front moved through in the afternoon that produced a line of severe thunderstorms that traversed the New York City metropolitan area. These storms produced microbursts with severe wind gusts locally in excess of 60 mph as documented in the Storm Prediction Center storm reports. As the storms traversed the areas, certain portions of the network were closed down due to both high winds associated with the thunderstorms and areas of heavy precipitation. This led to about 15% of trips at all altitudes between 1000 and 5000 ft AGL not being possible to complete between 1300–1500 local time (LT), given that convection closes down nodes at all of the corridor altitudes. Additionally, the cold front itself produced strong winds which further reduced the capacity at the higher altitudes (3000–5000 ft) between 1400–1700 LT. Given that the winds were generally weaker closer to the surface, there was rarely closure of nodes at lower altitudes due to strong winds for this day.

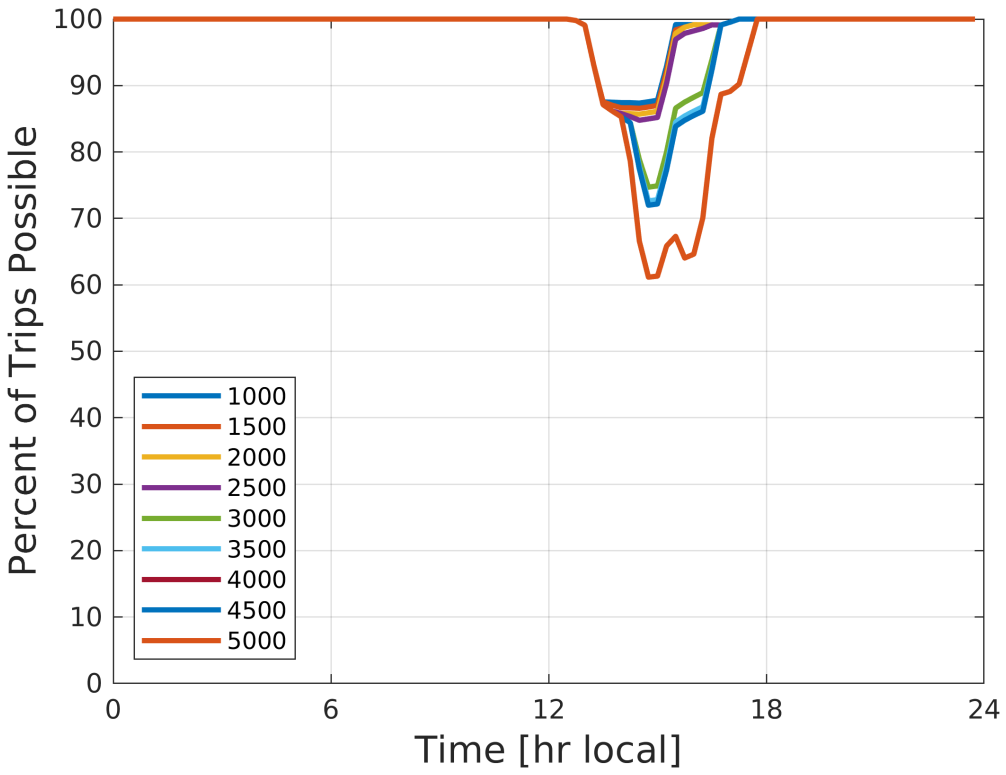

Figure 12. Percentage of scheduled flights possible to complete on 21 April 2020 at different corridor altitudes on the waterways network. The legend is for corridor altitude in ft AGL.

The second case day shown in Figure 13 is 12 August 2020. A sea breeze developed during the morning hours over the New York City area in response to the land areas warming up much more than the over the ocean. While this sea breeze was associated with generally light winds (less than 15 knots), it resulted in area of localized convergence over the city leading to the development of scattered pop-up thunderstorms. Where these thunderstorms developed and traversed, nodes were closed at all levels. This led to a decrease in the number of trips possible between 1200–1600 LT when these thunderstorms persisted. The decrease in capacity was at all levels, so there were no corridor altitudes that could be alternately used for some of the UAM traffic and some trips between origin and destinations were simply not possible if there was a thunderstorm along all possible routes.

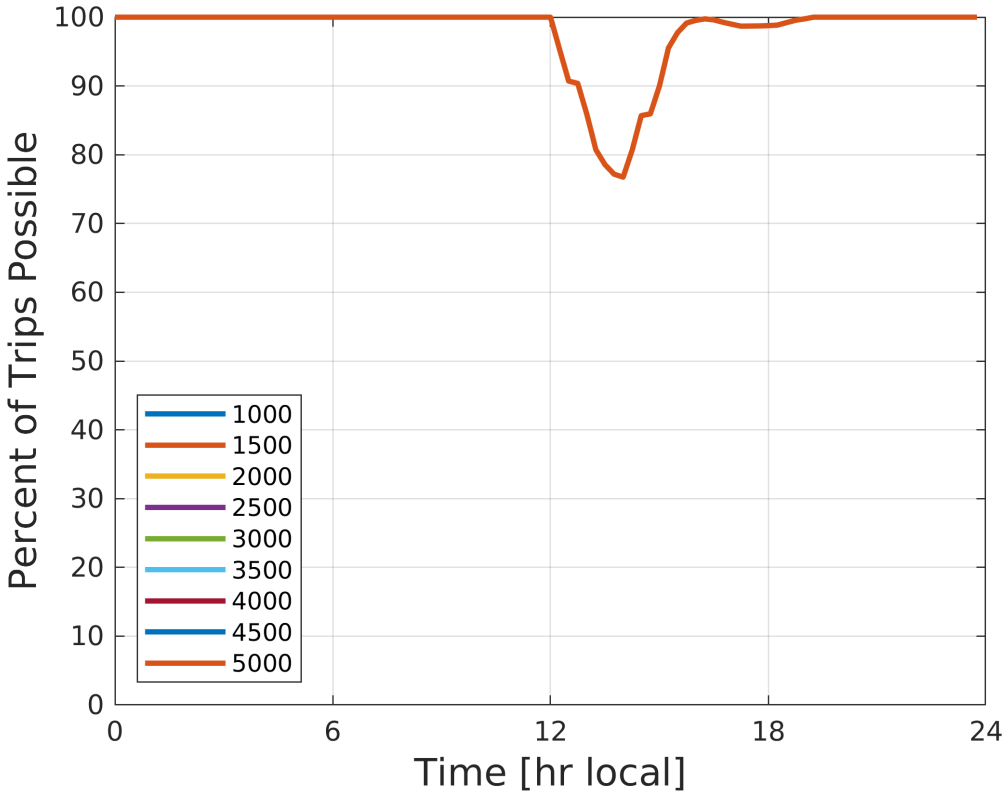

Figure 13. Percentage of scheduled flights possible to complete on 12 August 2020 at different corridor altitudes on the waterways network. The legend is for corridor altitude in ft AGL.

The third case day presented in Figure 14 is 26 March 2021. On this day, a strong low pressure system impacted the region resulting in strong winds and low clouds, including fog, for much of the day. As compared to the previous two cases, the strong winds on this day were generally synoptically-driven (e.g., large-scale) and persisted for many hours. Starting at around 0700 LT,

the number of scheduled trips possible at some of the higher altitudes (4500 and 5000 ft) began to plummet as the high winds overspread across the region. Over the next several hours, the lower altitudes were gradually impacted as winds continued to speed up, and with the underlying wind shear present the wind speed began to exceed the high wind threshold at lower altitudes. The peak impact was at 1100 LT when no flights were possible for corridors above 2500 ft, only 20% were possible at 2000 ft, and all flights could still be conducted at and under 1500 ft. Starting at around 1100 LT, the winds tended to lighten up and gradually more trips became possible until around 1500 LT when thunderstorms developed and moved across the network resulting in closed routes at all altitudes. Gradually, these storms cleared out and the number of routes open began to increase until 2100 LT when all routes were open and flights were possible for the remainder of the day.

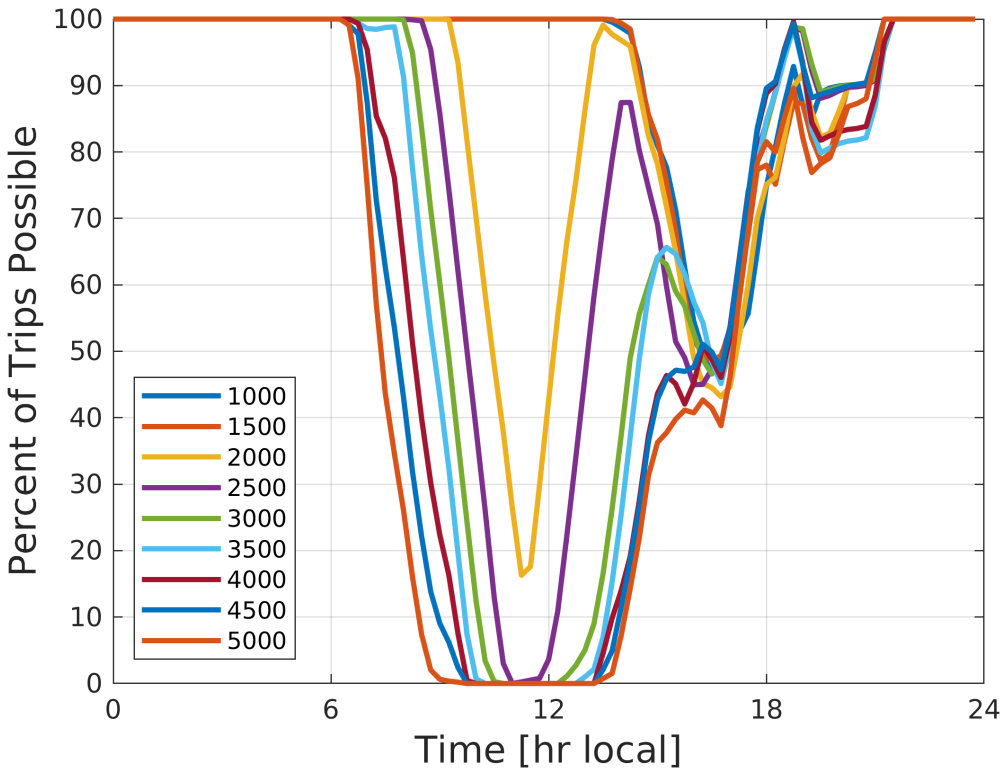

Figure 14. Percentage of scheduled flights possible to complete on 26 March 2021 at different corridor altitudes on the waterways network. The legend is for corridor altitude in ft AGL.

A summary of the impact of high winds and heavy precipitation (i.e., convection) on the successful trip completion for each case day can be deduced from Table 2 and Table 3. These tables respectively provide the percentage of scheduled trips that could be completed at 1000 ft and 5000 ft AGL. These altitudes were chosen because 1000 ft is generally where the impact is the least (most trips can be completed) and 5000 ft is where the impact is the greatest (least trips

TABLE 2

Percentage of Trips Not Possible at 1000 ft AGL

	Waterways	Helicopter	Direct
21 April 2020	1.1%	1.1%	1.2%
28 June 2020	1.8%	0.0%	0.5%
10 July 2020	8.6%	7.7%	4.0%
12 August 2020	2.0%	4.4%	1.9%
15 September 2020	0.0%	0.0%	0.0%
1 October 2020	0.0%	0.0%	0.0%
3 November 2020	0.0%	0.0%	0.0%
16 December 2020	0.1%	2.1%	0.6%
2 January 2021	0.0%	0.0%	0.0%
25 February 2021	0.0%	0.0%	0.0%
26 March 2021	7.0%	4.9%	2.9%
19 May 2021	0.0%	0.0%	0.0%

can be completed) due to the underlying shear generally present in the planetary boundary layer resulting in stronger winds at higher altitudes. This can be seen by the fact that the percentage of trips not possible for each corresponding network and day is always equal to or higher at 5000 ft than at 1000 ft by comparing Tables 2 and 3, a result of additional nodes being closed at higher altitudes due to high winds.

These tables also show the considerable variability in weather impacts depending on the conditions on any given day. Days with fair weather conditions, such as 15 September, 1 October, and 19 May are permissible for operations at all corridor altitudes for the entire day. On other days, the synoptic conditions result in widespread closures particularly at higher altitudes for much of the day (26 March, 16 December), generally in the winter or spring. Some days, local mesoscale conditions and thunderstorms result in intermittent route closures at all altitudes (28 June, 10 July, 12 August), particularly in the summertime.

On most days, each of the networks tested is affected very similarly. One result that is counter-intuitive is that the direct network design, where corridors only exist directly between the origin and destination vertiports, is the most resilient to adverse weather conditions. This was unexpected, as this network design only had one possible route for each scheduled flight while the other networks generally had alternative routes that could be possible if the shortest path was closed from hazardous weather conditions. The reason for this unexpected result is not obvious, but it is likely due to the fact that both the waterways and helicopter network have many of the main corridors over bodies of water which often have higher wind speed conditions compared to over land due to reduced friction. This may result in higher winds over the waterways and helicopter network nodes and more closures. Additionally, both the helicopter and waterways networks always require

TABLE 3

Percentage of Trips Not Possible at 5000 ft AGL

	Waterways	Helicopter	Direct
21 April 2020	4.0%	4.0%	2.8%
28 June 2020	1.8%	0.0%	0.5%
10 July 2020	9.4%	7.7%	4.0%
12 August 2020	2.0%	4.4%	1.9%
15 September 2020	0.0%	0.0%	0.0%
1 October 2020	0.0%	0.0%	0.0%
3 November 2020	17.53%	28.1%	19.1%
16 December 2020	24.0%	23.7%	23.2%
2 January 2021	19.2%	19.4%	19.3%
25 February 2021	3.5%	3.9%	4.0%
26 March 2021	40.5%	42.4%	35.6%
19 May 2021	0.0%	0.0%	0.0%

a longer route being taken compared to the direct network design; thus, it is more likely that an aircraft would encounter hazardous weather conditions enroute.

3. FUTURE WORK

This tool is under use in support of a multi-year NASA University Led Initiative program integrating the simulation capabilities of the UAMToolKit together with the sUAS testbed that MIT Lincoln Laboratory maintains [26]. The strategy is to integrate strategic planning modules developed by partner universities, specifically MIT and UT Austin, into the UAMToolKit. In a current collaboration with MIT Sloan, the tool is being used in a simulation framework to test optimization of UAM flights that incorporate re-balancing needs and weather robustness. It is expected the new optimization module will distribute the load on high utility corridors in the network, as evident in [22]. As previously presented, a new allocated program, *AAMGym*, will be utilizing this tool as a scenario generation tool to train decision support systems.

This page intentionally left blank.

A BLUE SKY INTEGRATION

In preparation for upcoming research programs that will utilize this tool, the team created a translation script to create scenario files that can be used in BlueSky [27]. BlueSky is a discrete time simulator that allow for large number of aircraft to be simulated simultaneously with a decentralized or centralized logic. For this integration, the network class *blueSkyScenarioTools.py* contains an expanding list of commands to allow a user to take the recorded data files, or scheduled file created for UAMToolKit and translated into a scenario files. The current functions available as part of this tool are described in Tables A.4. Note some of these functions are depended on the user having a network object as an input.

TABLE A.4

Available Methods to Translate UAMToolKit into BlueSky

Function Name	Description
defineAllwpt	Defines all nodes in network object as waypoints that can be referenced by name when creating commands for BlueSky
createAC	Creates an aircraft at the specified location and time
AddOrigDest	add the origin waypoint and destination for auto pilot functions in BlueSky to function properly
heading	low level heading change calculation between two nodes defined in LLA.
alt2txt	function that can take the altitude and convert to flight levels (e.g. 1000ft converts to FL010
addwpt	add a waypoint for the aircraft route at specified location
at	command to specify a waypoint or location at which to trigger a specified command
delcmd	command to delete an aircraft

An example of how to use these commands is provided below in code, as well as a sample of a BlueSky scenario file and a snapshot of the simulation in BlueSky Figure A.15. With this capability UAMToolKit can be used as a scenario generator that can sweep any parametric assumption used by the different modules used inside UAMToolKit. This is particularly important for testing large scale detect and avoid technologies (e.g., ACAS X, TCAS). A similar project called *AAMGym*, which utilizes BlueSky as simulator, will use the variety of scenarios created to test various machine learning and artificial intelligence algorithms with various objectives in mind, such as separation assurance, emergency landing, etc.

```
# network modules required
import networks.blueSkyScenarioTools as bst
import networks.networkUtilities as nsx
import numpy as np
```

```
import pandas as pd
import json
import networkx as nx
import random
import datetime
from omegaconf import DictConfig
import hydra


def loadJSON(file):
    with open(file) as f:
        output = json.load(f)
    return output

# hydra yaml file required to run this if no default parameters are given
@hydra.main(config_path="conf", config_name="scenario_generation")
def generate_aam_gym_scenarios(cfg: DictConfig) -> None:

    simTripsFile = cfg.simTripsFile # simulated trips file from UAMToolKit
    spFile = cfg.spFile # network file with specific path for OD pair
    plFile = cfg.plFile # network file with path length of file
    netFile = cfg.netFile # network GML file

    calculateSpeeds = cfg.calculateSpeeds # calculate speed from files
    date_ = datetime.datetime(100, 1, 1, 0, 0, 0)
    NAlt = cfg.numberAltitudes # random number of altitudes to assign
    vptNum = cfg.vptNum # number of vertiports

    # load the path files
    spJson = loadJSON(spFile)

    # load the path length
    plJson = loadJSON(plFile)

    # load the network file
    net = nx.read_gml(netFile)

    # load the speeds of aircraft files
    acSpeed = {}
    acCreate = {}
    # Need to read in the csv with the actual operations (i.e. simulated_trips)
    simList = []

    # vector of altitudes
```

```
routeSelector = np.zeros((vptNum, NAlt))
flightAltitudes = np.linspace(1000, 10000, NAlt)
# open the scenario file
with open(cfg.outputDir + "AAMoutput.scn", "w") as scnFile:
    # define all the waypoints and give them the name as the node number of
        ↪ the
    # network file so that we just have to call the network file
    bst.defineAllwpt(net, scnFile)
    with open(simTripsFile, newline="") as csvfile:
        sf = pd.read_csv(csvfile, delimiter=",", header=0)
        sf = sf.sort_values(by=["Departure Time"])
        nodesPos = nsx.getALLNodePositions(net) # nodes latlon #
        for index, rFlight in sf.iterrows():
            # load important information about vehicles
            acid = str(int(float(rFlight["Vehicle"])))
            orig = str(int(float(rFlight["Orign"])))
            dest = str(int(float(rFlight["Destination"])))
            depT = int(float(rFlight["Departure Time"]))
            arrT = int(float(rFlight["Arrival Time"]))

            # start a time vetor to write the execution time in scenario
            time = date_ + datetime.timedelta(0, 0, 0, 0, depT)
            time_ = str(time.time())
            time = date_ + datetime.timedelta(0, 0, 0, 0, arrT)
            destTime_ = str(time.time())
            deltaT = (arrT - depT) / 60 # in hours

            # Need to find the path of this flight
            pathNodes = [str(i) for i in spJson[orig][dest]]

            # calculate the speed of aircraft
            if calculateSpeeds:
                distance = nsx.getTruePathLength(
                    net, pathNodes, metric=False
                ) # Nautical Miles
                acSpeed[int(acid)] = distance / deltaT
            else:
                acSpeed[int(acid)] = random.randint(cfg.speed[0], cfg.speed[1])

            # get the position of the origin and destination
            origPos = nodesPos[str(pathNodes[0])]
            destPos = nodesPos[str(pathNodes[-1])]

            # assign origin and destination based on the label of the node
```

```
origLabel = "AAMN" + orig
destLabel = "AAMN" + dest
acLabel = "UM" + str(acid)

# route selector
altIndex = (
    routeSelector[int(orig), :] - depT
).argmin() # asssign the next free altitude
routeSelector[int(orig), altIndex] = depT
flightAlt = flightAltitudes[altIndex] + round(
    (random.random() - 0.5) * 100
) # add some noise to the altitudes
scnFile.write("# begin of UM" + str(acid) + "\n")
scnFile.write(
    bst.createAC(
        time_,
        acLabel,
        acType="UAMX",
        altitude=flightAlt,
        pos=origLabel,
        speed=str(int(acSpeed[int(acid)])),
        heading=bst.heading(origPos, nodesPos[str(pathNodes[1])]),
    )
) # create ac
scnFile.write(
    bst.addOrigDest(time_, acLabel, wptLabel=origLabel, orig=True)
) # origin
scnFile.write(
    bst.addOrigDest(time_, acLabel, wptLabel=destLabel, orig=False)
) # destination

for pn in pathNodes:
    pos = nodesPos[str(pn)]
    wptLabel = "AAMN" + str(pn)
    if destPos == pos:
        scnFile.write(
            bst.addwpt(
                time_, wptLabel, acLabel, altitude=flightAlt, speed
                    ↪ ="0"
            )
        ) # dest wpt
        scnFile.write(
            bst.at(
                time_,
```

```
                            acLabel,
                            wptName=wptLabel,
                            cmd=" DO" + bst.delcmd(0, acLabel, stack=True),
                        )
                    )
                else:
                    scnFile.write(
                        bst.addwpt(
                            time_,
                            wptLabel,
                            acLabel,
                            altitude=flightAlt,
                            speed=str(int(acSpeed[int(acid)])),
                        )
                    ) # add wpt
            scnFile.write("# end of UM" + str(acid) + "\n")
        csvfile.close()
    scnFile.close()


if __name__ == "__main__":
    generate_aam_gym_scenarios()
```

An example of the output file of the previous algorithm is show below where aircraft UM14, a UAMX type aircraft, is created one minute into the simulation at waypoint AAMN2 with inital heading 322, altitude of 986 ft, and speed of 134 knots. The aircraft is then deleted when it arrives at its destination. Note in this scenario file the aircraft is created at its cruising phase and does not model the takeoff and landing, although this is something that BlueSky can handle. In Figure A.15, you can see various of these aircraft flying along their specified. Note in the figure we highlight the route that a specified aircraft is taking (magenta line and waypoints).

```
0:00:00>DEFWPT AAMN0,40.74053659054681,-73.86927331611282,VOR
0:00:00>DEFWPT AAMN1,40.82305124638285,-73.94738200052652,VOR
...
0:00:00>DEFWPT AAMN169,40.82643722157611,-74.07031412594391,VOR
# begin of UM14
00:01:00>CRE UM14 UAMX AAMN2 322.8459872005562 986.0 134
00:01:00>ORIG UM14 AAMN2
00:01:00>DEST UM14 AAMN21
00:01:00>ADDWPT UM14,AAMN2,986.0,134
00:01:00>ADDWPT UM14,AAMN62,986.0,134
00:01:00>ADDWPT UM14,AAMN63,986.0,134
00:01:00>ADDWPT UM14,AAMN21,986.0,0
00:01:00>UM14 AT AAMN21 DO DEL UM14
```

Figure A.15. Simulation of 100 aircraft operating routes that were simulated through UAMToolKit and translated into a BlueSky scenario.

REFERENCES

[1] A.C.B. L. E. Alvarez, J. C. Jones, "Advanced Air Mobility Assessment Framework: FY20 Homeland Protection and Air Traffic Control Technical Investment Program," *MIT Lincoln Laboratory Project Report TIP-145* (2021), URL git@llcad-github.llan.ll.mit.edu:lalvarez/UAMToolKit.git.

[2] A. Riva, "When helicopters landed on manhattan skyscrapers all the time," *The New York Times* (May 2016), URL https://learning.blogs.nytimes.com/2012/05/16/may-16-1977-helicopter-accident-on-top-of-pan-am-building-kills-five/.

[3] P.D. Vascik, R.J. Hansman, and N.S. Dunn, "Analysis of urban air mobility operational constraints," *Journal of Air Transportation* 26(4), 133–146 (2018), URL https://doi.org/10.2514/1.D0120.

[4] R. Goyal, "Urban air mobility (uam) market study," *Booz Allen Hamilton* (November 2018), URL https://ntrs.nasa.gov/citations/20190001472.

[5] D.P. Thipphavong, R. Apaza, B. Barmore, V. Battiste, B. Burian, Q. Dao, M. Feary, S. Go, K.H. Goodrich, J. Homola, H.R. Idris, P.H. Kopardekar, J.B. Lachter, N.A. Neogi, H.K. Ng, R.M. Oseguera-Lohr, M.D. Patterson, and S.A. Verma, *Urban Air Mobility Airspace Integration Concepts and Considerations*, URL https://arc.aiaa.org/doi/abs/10.2514/6.2018-3676.

[6] FAA, "Urban air mobility concept of operations v1.0," (June 2020), URL https://nari.arc.nasa.gov/sites/default/files/attachments/UAM_ConOps_v1.0.pdf.

[7] B.P. Hill, D. DeCarme, M. Metcalfe, C. Griffin, S. Wiggins, C. Metts, B. Bastedo, M.D. Patterson, and N.L. Mendonca, "Uam vision concept of operations (conops) uam maturity level (uml) 4," (2020).

[8] UBER, "Uber elevate: Fast-forwarding to a future of on-demand urban air transportation. uber technologies," (October 2016), URL https://www.uber.com/elevate.pdf.

[9] C.S. Wayne Johnson, "Observations from exploration of vtol urban air mobility designs." (October 2018), URL https://rotorcraft.arc.nasa.gov/Research/Programs/eVTOL_observations_Johnson_Silva_2018.pdf.

[10] B. Tanner, "Wisk and new zealand government to partner in world's first autonomous air taxi trial," (February 2020), URL https://wisk.aero/blog/transport-trial-to-help-cora-take-off/.

[11] C. Courtin, M.J. Burton, A. Yu, P. Butler, P.D. Vascik, and R.J. Hansman, *Feasibility Study of Short Takeoff and Landing Urban Air Mobility Vehicles using Geometric Programming*, URL https://arc.aiaa.org/doi/abs/10.2514/6.2018-4151.

[12] C. Silva, W.R. Johnson, E. Solis, M.D. Patterson, and K.R. Antcliff, *VTOL Urban Air Mobility Concept Vehicles for Technology Development*, URL https://arc.aiaa.org/doi/abs/10.2514/6.2018-3847.

[13] S. Vora, "Uber copter to offer flights from lower manhattan to j.f.k." the new york times, june 5, 2019." (June 2019), URL https://www.nytimes.com/2019/06/05/travel/uber-helicopter-nyc-jfk.html.

[14] Senate - Commerce, Science, and Transportation, "S.516 - Advanced Air Mobility Coordination and Leadership Act," (March 2021), URL https://www.congress.gov/bill/117th-congress/senate-bill/516.

[15] M. Feary, "A firt look at the evolution of flight crew requirements for emerging market aircraft," (June 2018), URL https://ntrs.nasa.gov/citations/20180005720.

[16] M. Xue, *Urban Air Mobility Conflict Resolution: Centralized or Decentralized?*, URL https://arc.aiaa.org/doi/abs/10.2514/6.2020-3192.

[17] NATA, "Urban air mobility considerations for vertiport operations," *National Air Transportation Association* (2019), URL https://www.nata.aero/assets/Site_18/files/GIA/NATA%20UAM%20White%20Paper%20-%20FINAL%20cb.pdf.

[18] P.D. Vascik and R.J. Hansman, *Scaling Constraints for Urban Air Mobility Operations: Air Traffic Control, Ground Infrastructure, and Noise*, URL https://arc.aiaa.org/doi/abs/10.2514/6.2018-3849.

[19] FAA, "NUAIR Lays Automation Groundwork for High-Density Vertiports," (March 2021), URL https://nuair.org/2021/03/24/nuair-lays-automation-groundwork-for-high-density-vertiports/.

[20] NASA, "Urban air mobility market study," (2018), URL https://www.nasa.gov/sites/default/files/atoms/files/uam-market-study-executive-summary-v2.pdf.

[21] L. Rayle, D. Dai, N. Chan, R. Cervero, and S. Shaheen, "Just a better taxi? a survey-based comparison of taxis, transit, and ridesourcing services in san francisco," *Transport Policy* 45, 168 – 178 (2016), URL http://www.sciencedirect.com/science/article/pii/S0967070X15300627.

[22] L.E. Alvarez, J.C. Jones, A. Bryan, and A.J. Weinert, *Demand and Capacity Modeling for Advanced Air Mobility*, URL https://arc.aiaa.org/doi/abs/10.2514/6.2021-2381.

[23] L.W. Kohlman and M.D. Patterson, *System-Level Urban Air Mobility Transportation Modeling and Determination of Energy-Related Constraints*, URL https://arc.aiaa.org/doi/abs/10.2514/6.2018-3677.

[24] National Center for Atmospheric Research, URL https://www.mmm.ucar.edu/weather-research-and-forecasting-model.

[25] M.D. McPartland and T.A. Bonin, "Mobile Capabilities for Micro-Meteorological Predictions: FY20 Homeland Protection and Air Traffic Control Technical Investment Program," *MIT Lincoln Laboratory Project Report TIP-146* (2021), URL https://www.ll.mit.edu/sites/default/files/publication/doc/

mobile-capabilities-micro-meteorological-predictions-fy20-mcpartland-tip-146.pdf.

[26] A.D.C.M.S.F.J.P.M.R. M.P. Owen, L.E. Alvarez, "DC3: Distributed Command, Control, and Collaboration for sUAS: FY19 HP/ATC/HADR Technical Investment Program," *MIT Lincoln Laboratory Project Report TIP-117* (2020).

[27] J.M. Hoekstra and J. Ellerbroek, "Bluesky atc simulator project: an open data and open source approach," in *Proceedings of the 7th International Conference on Research in Air Transportation*, FAA/Eurocontrol USA/Europe (2016), vol. 131, p. 132.

This page intentionally left blank.