Project Report
TIP-145

Advanced Air Mobility Assessment Framework: FY20 Homeland Protection and Air Traffic Control Technical Investment Program

L.E. Alvarez
J.C. Jones
A.C. Bryan

24 March 2021

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: FY20 Homeland Protection and Air Traffic Control Technical Investment Program

L.E. Alvarez
A.C. Bryan
Group 42

J.C. Jones
Group 43

Project Report TIP-145

24 March 2021

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

Lexington Massachusetts

This page intentionally left blank.

TABLE OF CONTENTS

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. A flexible framework was designed to assess the operational viability of these technologies and the sensitivity to a variety of assumptions. This framework is 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.

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) is focused on the 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 [1]. Recent advancements in technology has reawakened interest in this urban mode of transportation to operate within cities [2]. Market research shows a potential for millions of operations of unmanned and manned UAM aircraft at low altitude on an annual basis. This anticipated number of on-demand air traffic is expect to stress the U.S. National Airspace System in an unprecedented manner [3].

To safely integrate this large number of operations, government entities have supported testing and evaluation, and provide an initial stand on operating procedures of UAM technologies through efforts like the NASA AAM National Campaign [4] and the FAA Urban Air Mobility Concept of Operations [5]. In alignment with these initiatives, several entities have been implementing the UAM vision by engaging in trial operations, releasing operation concepts, and studying vehicle design requirements [6], [7], [8], [9], and [10]. 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 [11].

The UAM passenger traffic flow is expected, at first, 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 [12]. 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 [13]. In addition, unlike commercial air travel, the landing and takeoff locations (vertiports) of UAM aircraft will likely be built through private capital [14]. Challenges facing the design of these corridors range from environment 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 [15]. For the design of robust operations, government and industry must understand complex infrastructure interdependences of corridor network design, landing and takeoff site capacity (vertipads), schedule assurance, and maximum sustainable throughput of the system.

The proposed analysis framework will support development of infrastructure requirements to support UAM airspace integration. This report describes a framework for assessing large scale, realistic operations of UAM, and demonstrates this framework in the New York City metropolitan region, including the development of models to characterize operational challenges faced by the technology. This report does not focus on aircraft design, on-board safety systems, separation requirements between aircraft, and well clear definitions for UAM. This report explores the network structure and its accompanying fleet-to-vertiport ratios required to accommodate an average passenger delay below 15 minutes, as further delay would be expected to impact passenger willingness to travel [16], [17]. The rest of the report is organized as follows. Section II describes the methodology being followed in the framework development. Section III defines experimental design of our analyses which are planned, results of which will be reported in the full report.

2. MODELING

The simulation framework is composed of a demand model, a scheduler, and a discrete-event simulator. A demand model composed of Poisson distributions drives the traffic requests for UAM. This differs from previous studies that used a synthetic probability distribution to drive events [18]. A greedy scheduler assigns aircraft to the demand based on the state of the closest available aircraft. A discrete event simulator then ingests the scheduled traffic and creates a sequence of events that are reassessed each time step.

2.1 DATABASE

For use in the demand model, a database holding taxi and for-hire-vehicles (FHV) was created using public data from the New York City government website [19]. For our purposes, a PostgreSQL database called *uamdb* was created on the Lincoln Laboratory super-computing grid system (LLSC). A set of tools and instructions on how to to scrape and handle this data are provided by Todd Schneider on his github nyc-taxi-data [20]. Using these tools, the raw data was downloaded to the LLSC and added into the database. Separate tables were created for FHV and taxis. Additionally a set of tables, one for each hour of the day named "trips_hour_px" for hour between 0 and 23, was created with only desired fields inserted. An example of one entry from the table holding the trips from 00:00 to 00:59 is given in Table 1.

TABLE 1

Trips at hour 00

Field	Data
cab_type	green
pickup_datetime	2015-10-05 00:00:00
pickup_longitude	-73.910957336425781
pickup_latitude	40.776077270507812
dropoff_datetime	2015-10-05 00:06:00
dropoff_longitude	-73.930618286132813
dropoff_latitude	40.764198303222656
passenger_count	1

Currently, the "trips" table which holds all the trips from 2009 to 2019 for green and yellow taxis has 1,193,983,722 total entries with many of these entries having greater than 1 passenger per trip. The number of trips per day is anywhere from 300,000 to 1,200,000 with the average per year increasing over time. In addition to the PostgreSQL database within LLSC, the program uses a shared directory *UAM_shared* to house data and Jupyter notebooks with algorithms that are used for pre-processing and post-processing of the data. Jupyter notebooks allow the program to interactively query the database as well as develop algorithms prior to submission to the main github repository lalvarez/UAMToolKit [21]. The Python package psycopg2 was used to query the database. Example scripts can be found in UAM_shared/database_setup. The LLSC can also be used to load an interactive shell to query the database. To do this, from a login or compute node, change directory to /usr/local/CloudStack/scripts-v4 and run "./db_shell txg-uam_database". From here you can run "\connect uamdb" to connect to the database where you can now run and view sql queries in the console. In the future, creating more pre-processing scripts to ensure data reliability, and only creating tables with the data that will get used, will speed up the queries and prevent from needing to recreate tables if problems are found.

2.2 MODELING DEMAND FOR URBAN AIR MOBILITY

As described in Section 2.1 the LLSC was used to maintain the large quantity of data within a PostgreSQL database and shared directory drives. This database contains taxi requests from 2009 to 2019 for yellow cab, green cab, and vehicle for hire (e.g., limousine, Uber, Lyft) services with a pickup and drop-off location in the metropolitan region of New York City (NYC) [19]. For this report the vehicle-for-hire data was ignored due to its inconsistent quantity of pickup and drop-off locations. Using the New York Taxi Database, a demand model was developed to capture the temporal and spatial variations of potential UAM demand.

2.2.1 Demand Model Creation

Although the underlying data for this model is ground-based, UAM traffic is expected to have differences in demand based on the time of the day, and day of the week. This report assumes 5% of the requests accross all days and times are attributed to UAM traffic as it is expected initially to be too expensive for the general public. This assumption also aligns with market studies from NASA [3]. An example of the raw taxi weekday demand is shown by Figure 1. From Figure 1 it is observed that demand does not dropoff after the initial morning demand. After further investigation it was determined that the steady demand between 0800 and 1500 was in fact realistic as it compares with traffic information found in literature. Similar demand data is available for the weekend, but for the purpose of this report we focus on the weekday demand.

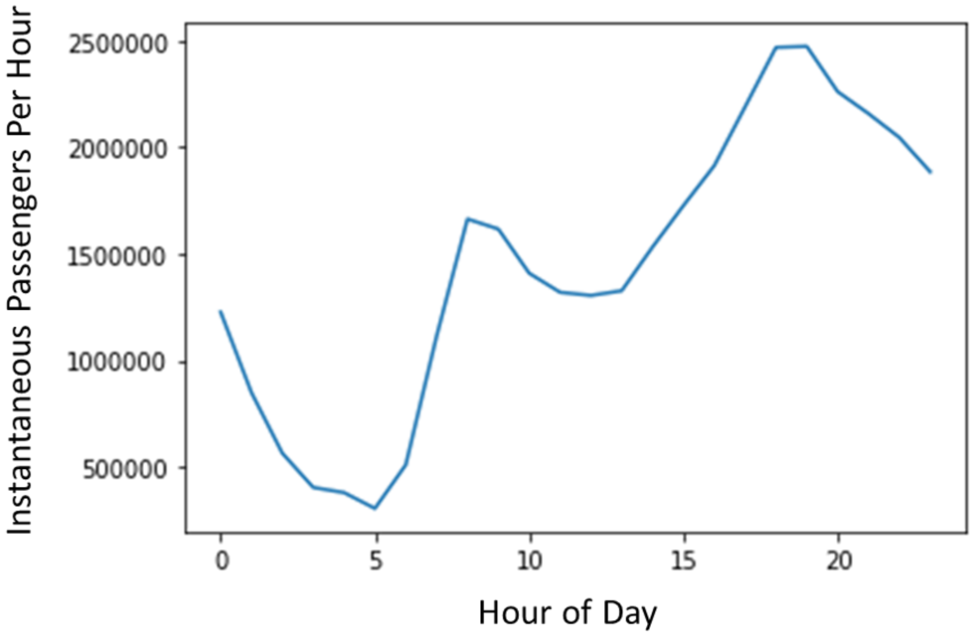

Figure 1. Demand throughout day from taxi database.

Prior to 2017 the database contains the exact latitude and longitude of the request's origin and destination. For the purposes of this report we show the data used from training the demand model on all of the weekday data for 2016. As shown in Figure 2 the database has enough spatial resolution that the model is able to accurately predict the destination of the demand for any time of the day.

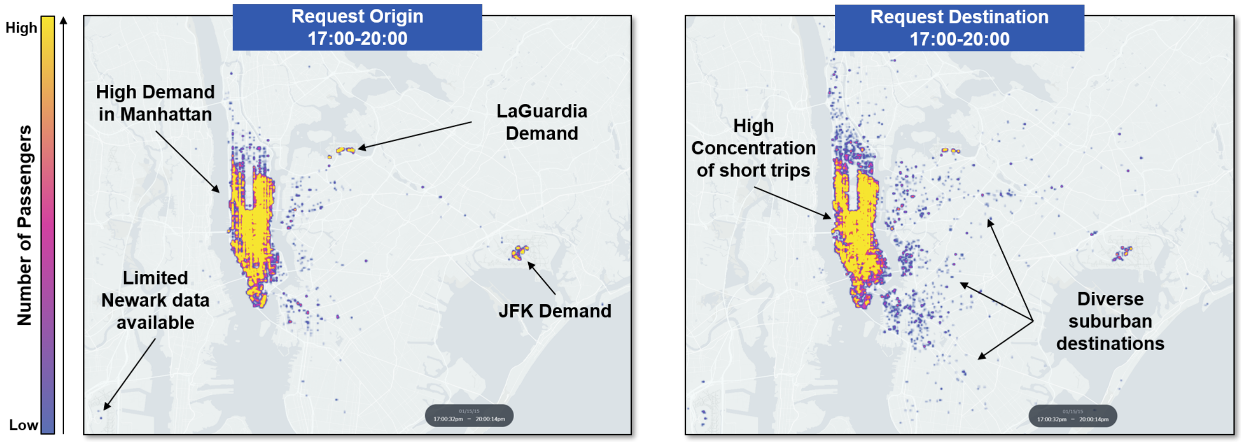

Figure 2. Example Origin of demand (left) and the destination (right) distributions of demand between 1700 - 2000 for a typical weekday.

A Jupyter notebook named *demand_connected.ipynb* contains the general algorithm used to create the large lookup tables containing the distributions that are used for each hour requested by the user. To simplify data processing, each year of data is sorted into the seven weekdays and the number of requests are grouped based on the closest discretized bin. The LLSC Jupyter notebook interface is used to process the data and is used to request an exclusive compute blade, specifically one with RAM greater than 128GB. The complete process of creating a full hour's look up table takes approximately 30 minutes, making a full day's worth of lookup tables a 12 hour process. The lookup tables are contained in the shared LLSC directory *UAM_shared/kvalues/* and are approximately 100MB each. A general structure of the algorithm used to create the lookup tables is shown below:

Algorithm 1 Database Look Up Table Creation

```
function CreateLookUpTables($latitudesOfRegion, LongitudeOfRegion, HoursOfInterst$)
▷ LatitudeOfRegion and LongitudeOfRegion define the bounding box of interest
    $pk_{dic}$ dictionary of origins
    while $hour \leq max(HoursOfInterest)$ do
        $Select(pk_{lat}, pk_{lon}, dp_{lat}, dp_{lon}, cab_{type}) \leftarrow Database_{year}$
        for $pk_{pairs}$ to $All$ do
            if $cab_{type}$ =Taxi then
                $\lambda = histogram(dp_{lat}, dp_{lon})$
                $pk_{dic}(pk_{p}airs) = \lambda$
            end if
        end for
        $save(pk_{dic})$
    end while
end function
```

At its core, the demand model uses a Poisson distribution to represent the stochastic UAM requests. Literature has commonly used Poisson distributions to represent demand from a variety of resources as it is has a well-known predictable output and reduced computational complexity [22]. Thus, the previously saved distributions from Algorithm 1, are are used as the λ parameter for the Poisson distribution to calculate the number of passengers at each bin. To maintain a fast time simulation framework with simulation times on the order 20 minutes, the NYC metropolitan region is discretized into 100 m^2 areas to create sets of 360x720 grid worlds that encompass the bounded area by these latitude and longitude pairs (40.54981, -74.26003) (40.8751,-73.5363). In order to preserve the spatial relation between a requests origin-destination (OD) pair, the distributions are grouped based on the origination. For each hour of the day a 360x720 origin grid world is created where each bin contains an equivalent 360x720 grid of possible destinations, for a total of 67 billion pairs.

To reduce the demand model run time, the origin and destination grid are represented by a sparse matrix where non-zero entries are those where the NYC database shows a request existed. This representation removes grid bins over water and regions outside of the NYC metropolitan database. During the first year of the program it was settled that the time resolution of the framework would be on the order of 1 minute, therefore λ represents the frequency of passenger on a per minute basis. This lambda parameter is then used in simulation to sample from a Poisson distribution for the specified time span desired at a specified destination bin. The sum of all random samples are then attributed to the origination bin. A general depiction of the accumulation of demand from the destination gridworld to the origin gridworld is shown in Figure 3. The bins of a destination grid contain the number of requests observed in the time scale desired (represented by λ).

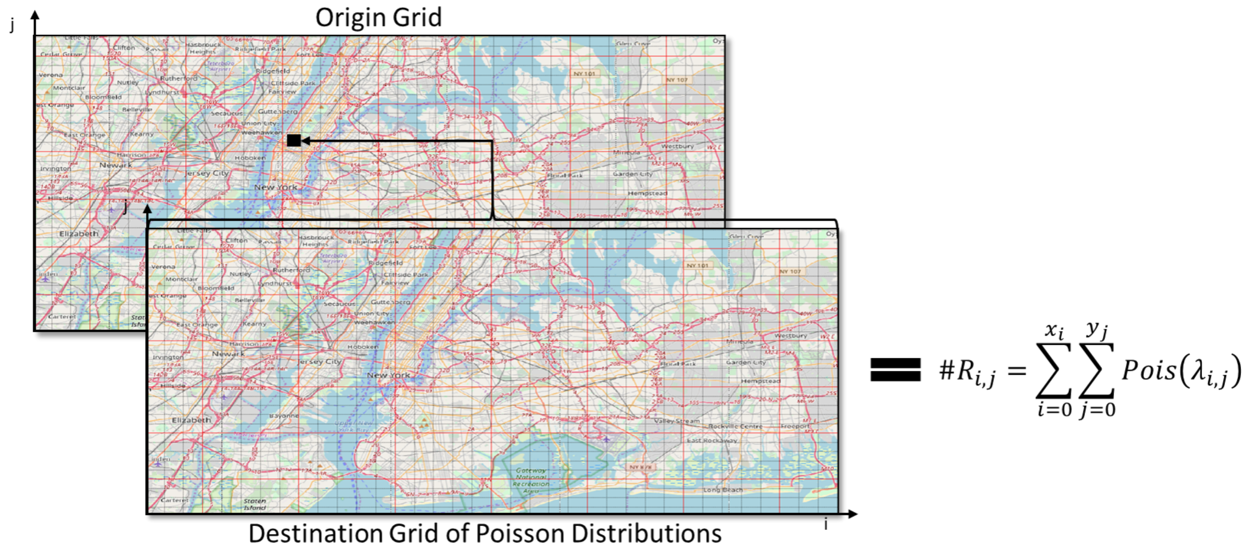
Origin Grid
j
j
Destination Grid of Poisson Distributions
i
$\#R_{i,j} = \sum_{i=0}^{x_i} \sum_{j=0}^{y_j} Pois(\lambda_{i,j})$

Figure 3. 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.

2.2.2 Vertiport Locations

The demand model outputs the raw location of the requested UAM demand, this request is the associated with the nearest vertiport location in our network. Since the initial vertiport structures are expected to leverage existing heliports, the model assumes public heliports as part of the vertiport network. In 2019 there were three public heliports operating in the Manhattan region, see Figure 4. In addition to these locations, existing heliports were available at the major airports in the New York metropolitan region (i.e., John F Kennedy International, La Guardia, and Newark Liberty International). Although Teterboro airport is in the vicinity there was no taxi traffic captured by the NYC taxi database to and from this destination. These four vertiports where initially tested as part of our network and were quickly determined to be an insufficient quantity for demand, requiring hundred of more than 80 parking spots per vertiport to accommodate demand.

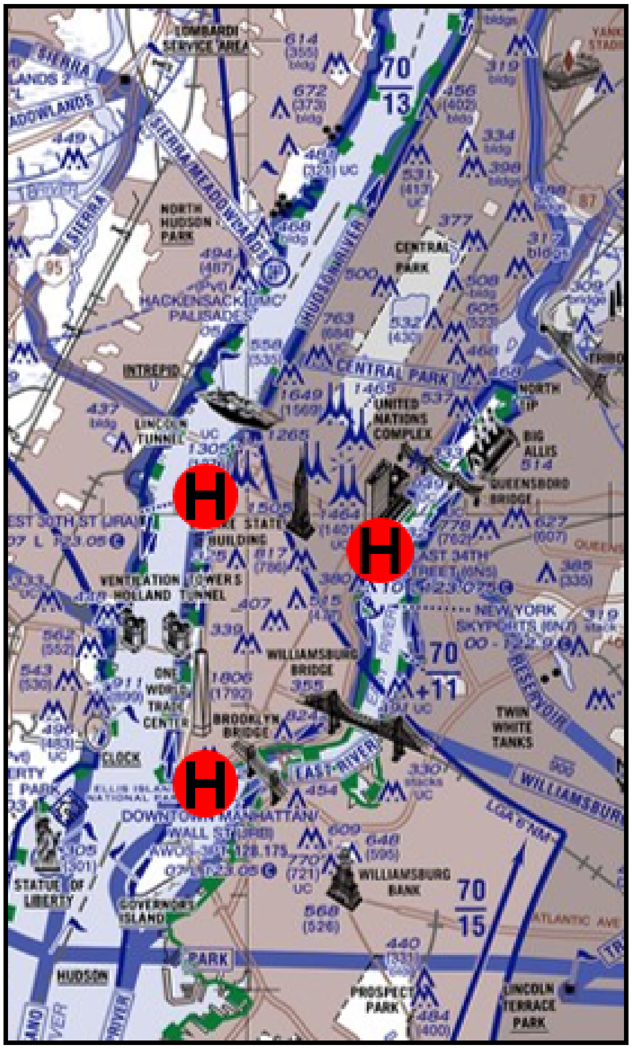

Figure 4. Public Manhattan heliports with operations in 2019.

K-means clustering was then used to find new vertiport location that would meet the demand as efficiently as possible. In an attempt to reduce overfitting the data, the original vertiport locations were use in the K-means clustering and additional vertiport locations were explored. A full day of demand was used as the data to cluster. A sweep from 6 to 30 possible clusters was performed using the k-means algorithm from the Python package *sklearn.cluster*. Plotting the inertia of all clusters showed a clear knee in the fitting criteria at 29 vertiports. The cluster latitude and longitudes were then inspected and moved to the nearby structures or parks that could support the operations. An example of the vertiport location chosen and how demand would be divided throughout the day is shown in Figure 5.

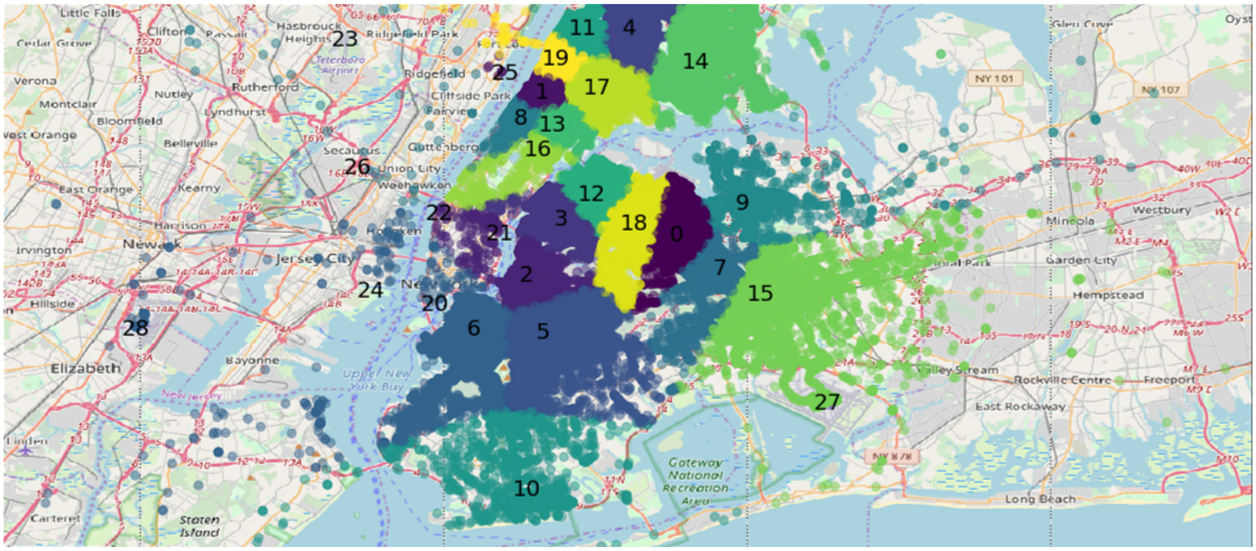

Figure 5. A days worth of demand assigned to the final vertiport locations provided from K-means clustering and existing public heliport locations.

The symmetry in origin and destination grid worlds allows each bin to be assigned to the nearest vertiport separate from the demand processing. The number of vertiports and locations can initially be chosen based on geographic regions close to waterways that could support the infrastructure, however further analysis using K-means clustering of taxi data is also presented in this report. During simulation run time, the demand model expects a specified time window, starting day of the week, and the time scale desired for the output. For example, the traffic associated with an OD pair can be defined as separate events for every minute of simulation or as single events occurring in an hour. At run time, the demand model collects the total number of requests for an OD pair with its associated time stamps. This information is then recorded and sent to the scheduling module as a set of events.

2.3 AIRCRAFT SCHEDULING

The scheduling component of the model outputs a list of flights, throughout a set period of time, that satisfy the given demand model. To accomplish this, the scheduler iterates through the demand on a per minute basis assigning demand to aircraft until either all aircraft are full or there is nothing left to assign. The scheduling function inputs are described in Table 3.

TABLE 2

Scheduler Function Input

Input	Data Type	Description
vertiport_path	string	filepath to the .npy with the vertiport lat/longs
numVehicles	int	number of operating aircraft
capacity	int	how many people per aircraft
speed	int	aircraft speed in kts
demOutput	string	filepath to csv output from demand model
timeHorizon	int	how many hours to loop for
timeres	string	’min’ or ’hour’ tells you the precision from demand model.
direct	boolean	direct flight paths or custom
short_path_path	string	filepath to json holding pairwise distance between nodes
random	string	the type of distribution for the initial aircraft placement
turnAround	int	how long a helicopter would wait before taking off again
steadyCalc	boolean	true if assuming no takeoff and landing dyanamics
nodeBuilder	boolean	if you are using output from node_builder tool or not

The output is a list of trips consisting of aircraft identification, departure time, time of arrival, departure vertiport, and arrival vertiport An example of the first few lines of an output file is given in Table 3.

TABLE 3

Scheduler Function Output

Aircraft Id	Current Vertiport	Planned Vertiport	Departure Time	Arrival Time	Passengers
10	22	9	0	19	1
18	22	1	1	16	1
23	22	5	1	11	2
31	22	9	1	20	1
12	16	9	2	20	1
252	22	1	2	17	1

For testing purposes, there is a copy of the scheduler in a Jupyter notebook in the shared directory UAM_shared\scheduler called newSched_test.ipynb. There are also tools for plotting the per person wait times in the same notebook. Typically more than one schedule is run at a time. To accomplish this, the slurm capabilities of the LLSC are used. The scheduler folder mentioned above contains the script schedule_batches.py. This script can specify an array holding the different number of aircraft to use, an array of the different capacities, and the parameters for the scheduler. Running the script will launch every pairwise combination of aircraft number and capacity as a grid job.

Areas not addressed by this module are the number of vertipads at each vertiport, flight separation standards, and turnover rates at vertiports. However, there is an inherent one minute time period between landing and takeoff (e.g., minimum time step). Theoretically, every aircraft could simultaneously be assigned to fly to one vertiport, from the same vertiport, at the same time, as long as it followed the demand model. The scheduling module inputs consist of the number of aircraft, aircraft cruising speed, aircraft-vertiport distribution, demand data, and the travel times between vertiports. The aircraft-vertiport distribution indicates the starting vertiports for each aircraft, a uniform distribution being the default. All aircraft start in state zero (awaiting assignment) with a total state set of five; the states are defined in Table 1.

TABLE 4

Aircraft State Sets Available in Scheduling Module

Aircraft State	Description
0	aircraft at vertiport with no demand assigned
1	aircraft assigned to pick up demand at a different vertiport but not yet flying string
2	aircraft assigned to transport demand from current vertiport to another but not yet flying
3	aircraft assigned to pick up demand at different vertiport and flying
4	aircraft assigned to transport demand from current vertiport to another and flying

Demand is considered up to the current time step with a given resolution and is not forward looking. Each vertiport has a weighting factor used when scheduling, such that priority is given to the vertiport with the highest weight: it is equal to the summation of the accumulated delay of all the passengers waiting at that vertiport. The first strategy involves assigning aircraft in state zero, one, or two, where if in state one or two the requests must have the same destination as the aircraft. When multiple aircraft are found in this state the aircraft with the most open seats are selected. The second strategy prioritizes aircraft in state one or three that are scheduled to pick up demand at the vertiport of interest and that have the same destination. Aircraft in state one assigned in this way will also take any available demand from its current vertiport to the vertiport that it is assigned to pick up from, regardless of the weight of its departure vertiport. If the aircraft is in state three, it will simply pick up the demand once it has arrived. In the case where there are no aircraft able to satisfy the demand, the demand is held to be checked again at the next time step (i.e. the next minute of simulation time). Parties of people are not taken into consideration when checking demand. For example, if there are four requests at the same time at a vertiport, and there are four aircraft with one open seat each, these four people will be split among the aircraft.

This example only holds true if the aircraft are headed to the same destination. However, if there are four open seats in one aircraft, all four requests will be assigned to this single aircraft. Future implementation are expected to maintain parties together. An example scenario involving a system of two vertiports and a fleet of four aircraft is shown in Figure 2. Once an aircraft is assigned, its travel time is calculated. By default, this is done by using the pre-calculated corridor distances between the vertiports, and dynamics simulating takeoff, landing, and cruising, where the takeoff and landing profiles are the same but reversed. As an example, in our 15 vertiport system, the mean corridor distances between vertiports is 18 nautical miles (nmi), with a maximum of 41.8 nmi and a minimum of 3.1 nmi. Direct flight paths between vertiports can also be used to calculate travel time and this is done within the scheduler. After assigning all possible demand for a time step, aircraft states are updated.

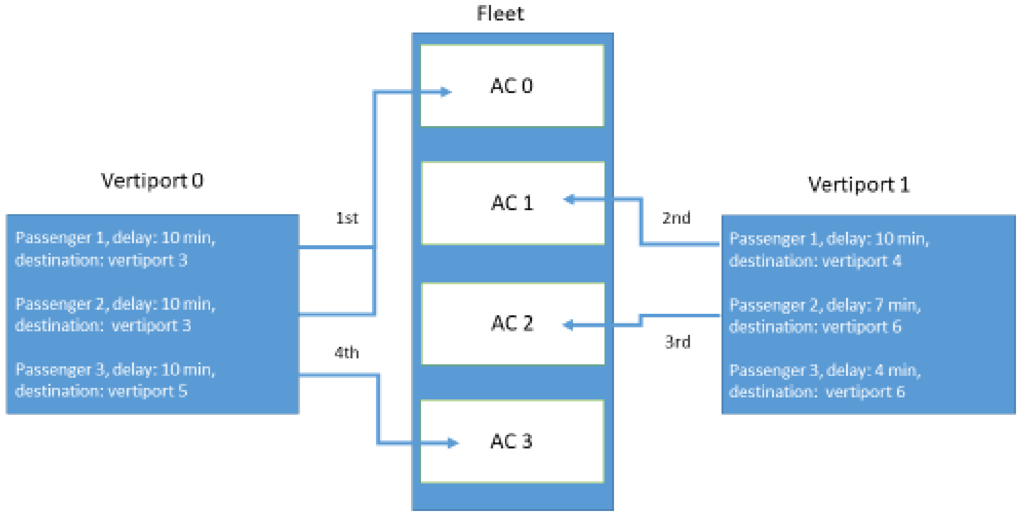
Fleet
AC 0
AC 1
AC 2
AC 3
Vertiport 0
Passenger 1, delay: 10 min, destination: vertiport 3
Passenger 2, delay: 10 min, destination: vertiport 3
Passenger 3, delay: 10 min, destination: vertiport 5
1st
4th
Vertiport 1
Passenger 1, delay: 10 min, destination: vertiport 4
Passenger 2, delay: 7 min, destination: vertiport 6
Passenger 3, delay: 4 min, destination: vertiport 6
2nd
3rd

Figure 6. Demand assignment.

2.4 NETWORK DESIGN

To calculate distance between vertiports, either direct flight paths between vertiports, Figure 7, or a system of nodes and edges that represent flight paths is used. An application was created to allow creation of custom flight paths given a metropolitan region map, bounding box, and vertiport coordinates in latitude and longitude. When finished, it outputs the travel times to be read by the scheduler. Figure 8 shows a work in progress of a node network on the top and a completed version on the bottom. Red nodes represent vertiports while blue nodes represent the nodes along the flight path. Every node is numbered and edges between any two nodes can be added at any time. When individual nodes are drawn in succession they are automatically connected by an edge. These nodes are temporary and can be removed before being locked in. Whenever a set of nodes and edges is locked in, a checkpoint is created that can be used to load previous configurations. These checkpoints can also be loaded after closing and reopening the application. After the system has been created it uses Dijkstras algorithm [23] to find the shortest path between every set of vertiports.

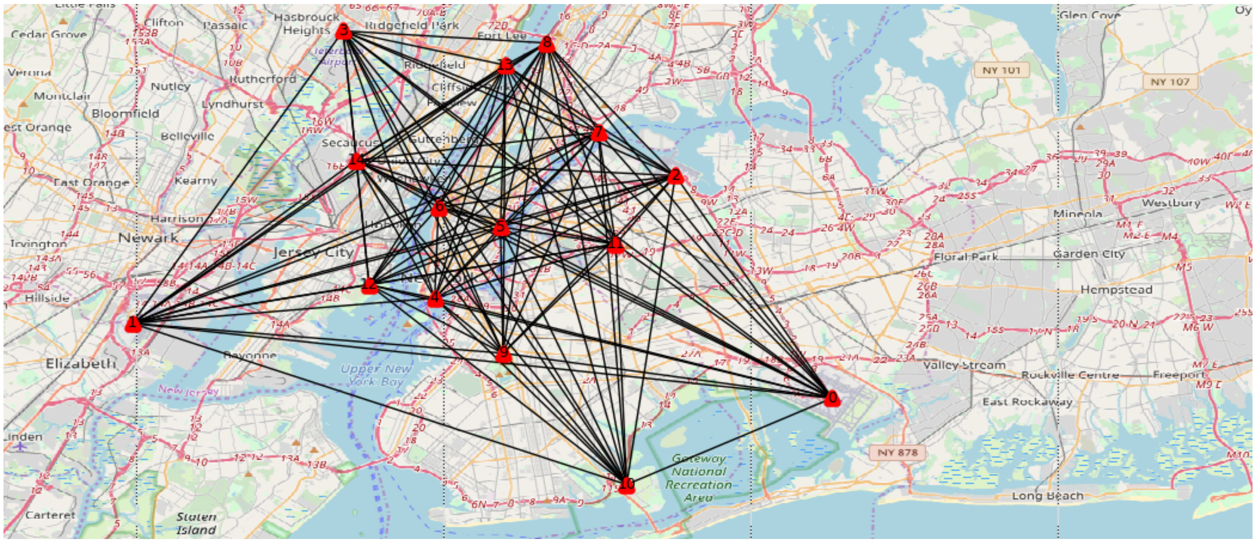

Figure 7. Direct travel corridor network.

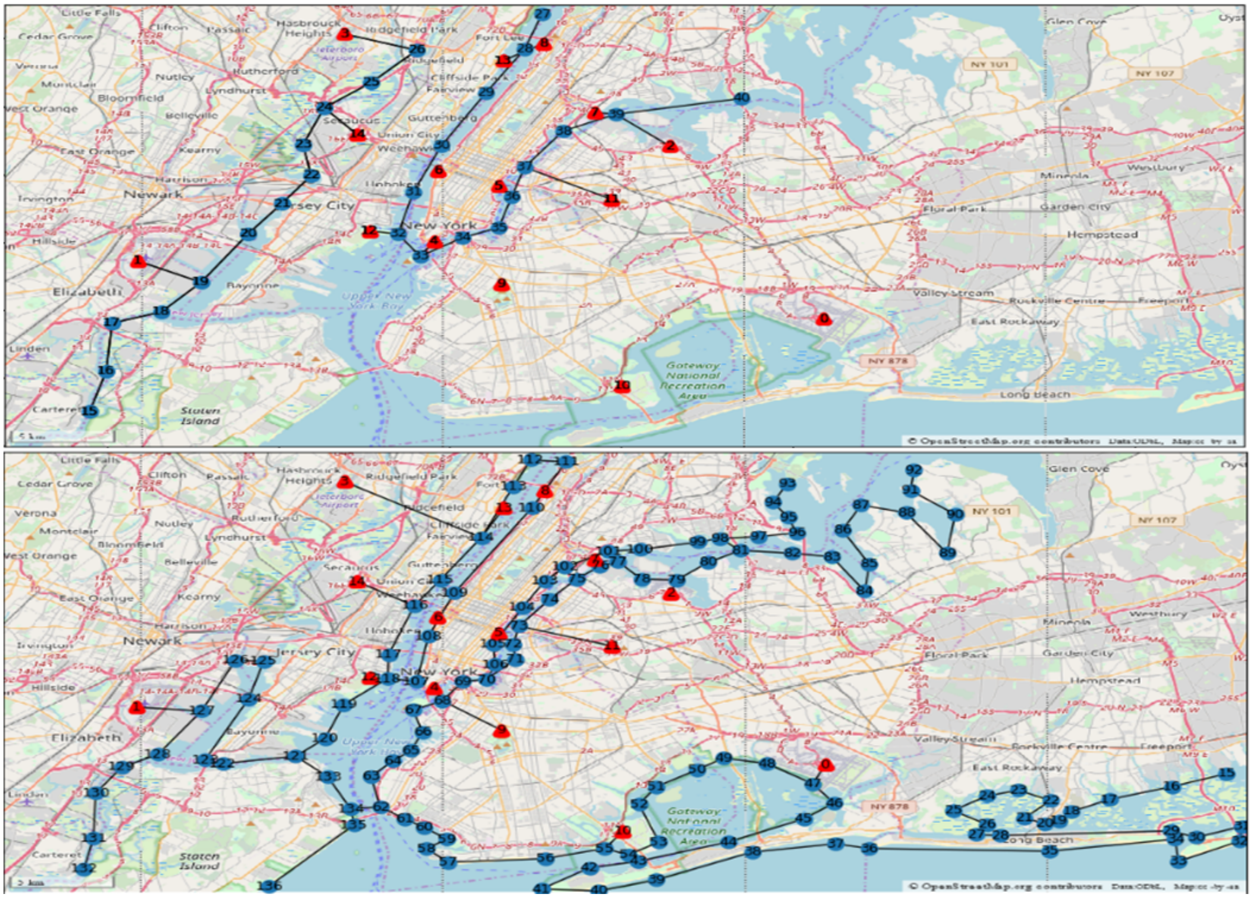

Figure 8. Corridor and vertiport network creation in progress (top) and completion (bottom).

The initial design of UAM routes is expected to be a merger between Visual Flight Rules (VFR) and Instrument Flight Rules (IFR) routes, account for noise abatement, and account for population density. As previously mentioned, in 2019 the FAA released their Concept of Operation V1.0 [5], within this document the FAA highlights a set of simple corridors. In this CONOPS the FAA expects that within UAM corridors:

- All aircraft operate under UAM specific rules, procedures, performance requirements, and surveillance requirements
- Fixed wing aircraft and UTM aircraft (i.e., small UAS opeartions) may cross UAM corridors
- Helicopters and UAM aircraft may use or cross corridors
- UAM corridors do not vary with airspace class (i.e., corridors may traverse Class G, C, D, E, B airspace)

From this criteria the corridor design tool previously described was used to create a network structure that follows the VFR routes in the helicopter sectional for New York, example except shown in Figure 9. However, to be conservative on noise abatement measures, and population density, the UAM corridors where structured to traverse over water, and only over land to directly reach the vertiport from the shortest path to the water, Figure 8 (bottom). Note: the initial corridor design did not account for altitude offsets and is expected to fly at 1,000 ft MSL. The file *vertiport_29.npy* contains all of the vertiport locations for the network, while *latLongDict_29.npy* is a dictionary mapping the network nodes to their respective latitude and longitudes. A helper file is also available which maps the shortest path along the nodes from one vertiport to another (*shortestPaths_29.json*). A companion to this mapping file is *shortestPathsLen_29.json* which outputs the measured distance along this shortest path in nautical miles.

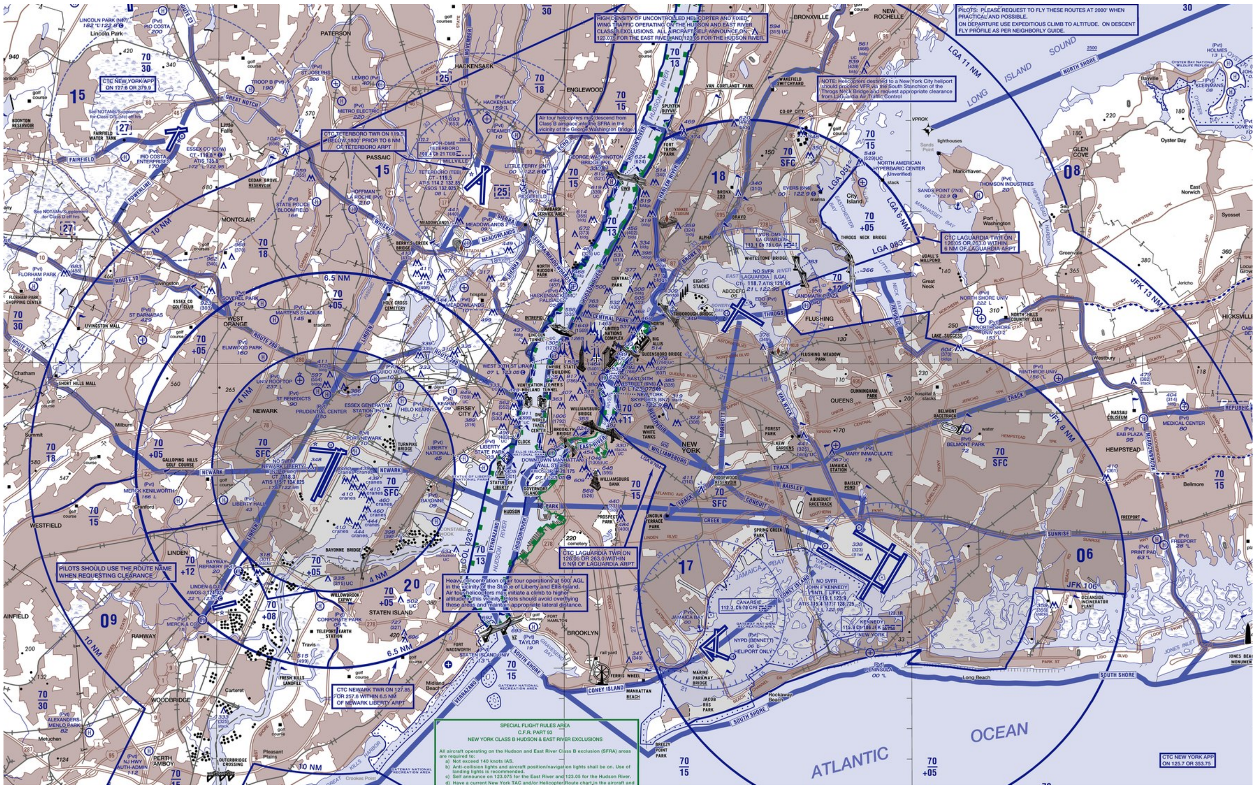
PATERSON
HACKENSACK
ENGLEWOOD
BRONXVILLE
NEW ROCHELLE
LONG ISLAND SOUND
NORTH SHORE
LGA 11 NM
LGA 6 NM
LGA 083
JFK 13 NM
JFK 8 NM
JFK 106
PASSAIC
MONTCLAIR
WEST ORANGE
NEWARK
JERSEY CITY
STATEN ISLAND
BROOKLYN
QUEENS
FLUSHING
NEW YORK
HEMPSTEAD
WESTFIELD
LINDEN
RAHWAY
WOODBRIDGE
CONEY ISLAND
ATLANTIC OCEAN
SOUTH SHORE
6.5 NM
4 NM
10 NM
SFC
PILOTS SHOULD USE THE ROUTE NAME WHEN REQUESTING CLEARANCE
CTC NEWARK TWR ON 127.85 FOR 257.8 WITHIN 6.5 NM OF NEWARK LIBERTY ARPT
CTC LAGUARDIA TWR ON 126.05 OR 263.0 WITHIN 6 NM OF LAGUARDIA ARPT
CTC NEW YORK APP ON 125.7 OR 353.75
CTC TETERBORO TWR ON 119.5 BELOW 1500' PRIOR TO 6 NM OF TETERBORO ARPT
PILOTS: PLEASE REQUEST TO FLY THESE ROUTES AT 2000' WHEN PRACTICAL AND POSSIBLE.
SPECIAL FLIGHT RULES AREA
NEW YORK CLASS B HUDSON & EAST RIVER EXCLUSIONS

Figure 9. Excerpt from section showing the helicopter routes in the New York metropolitan region.

2.5 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 section II B. 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 (with range 5 10 minutes). 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 10.

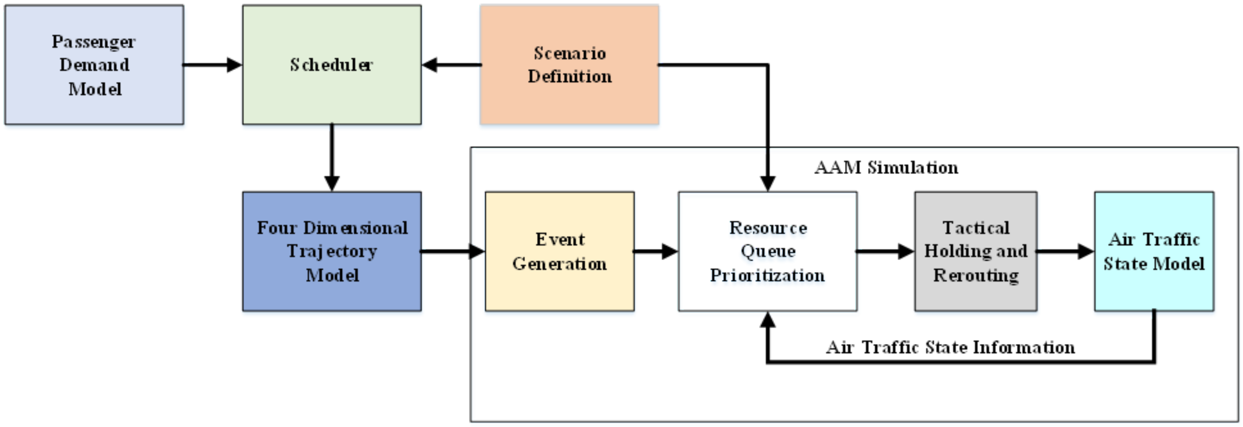
Passenger Demand Model
Scheduler
Scenario Definition
AAM Simulation
Four Dimensional Trajectory Model
Event Generation
Resource Queue Prioritization
Tactical Holding and Rerouting
Air Traffic State Model
Air Traffic State Information

Figure 10. 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 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.

3. RESULTS

In year one of the AAM Assessment Framework program we set out to validate NASA's 2030 forecast which expects 10,000s of UAM operations over a metropolitan region. The forecast is based on UAM operation replacing 5% of the ground transportation. Note in the initial simulations it is assumed the aircraft fleet has homogeneous dynamic capabilities and there's no competing entities, that is to say the entire UAM fleet operates under one scheduling system. The aircraft dynamics assumed for the simulations are presented in Table 5.

TABLE 5

UAM Aircraft Assumptions

Aircraft Property	Minimum Value	Maximum Value	Notes
Cruising Altitude	1,000 ft	1,000 ft	NASA working groups, and FAA conversation led to initial assumption of UAM operations below 1,200 ft
Cruising Speed	50 knots	80 knots	Cruising speed derived from conversations with SMEs, A^3, Uber, Wisk
Landing & Takeoff Speed	0 knots	50 knots	Although aircraft are expect to have vertical takeoff capability SME forum expected 10 knots to be a reasonable forward speed during takeoff and expects aircraft accelerate up to cruising speed
Vertical Rate	500 ft/min	500 ft/min	Conservative value chosen from helicopter radar data
Passenger Capacity	4	10	Sweep is used for fleet size requirements, but industry is converging to 4 passenger aircraft

The relationship between fleet size and average passenger delay was explored with various vehicle passenger capacities. In this analysis the vertiports where expected to have enough vertipads to accommodate a uniform distribution of aircraft ready for takeoff at the beginning of the simulation. For example, using the 29 vertiport network, a fleet size of 2900 aircraft would require 100 vertipads per vertiport. The authors are aware that it is likely unfeasible to have vertiport structures that support this number of aircraft and future work explains extended assumptions that could be explored. This analysis held constant the demand by using the same random seed, and repeating the daily demand experienced. As seen in Figure 11, two behaviours can be are confirmed: 1) an equivalent average wait time per person can be met by a fleet composed of vehicles with higher passenger capacity, 2) increasing fleet size has marginal benefits in reducing the average wait time per person.

Fleet size and experienced delay have an almost linear relationship up to 100 minute delay regardless of vehicle passenger capacity. Of relevance the results show a 40% reduction in the fleet size required to meet the same average delay between a four passenger vehicle, which industry is converging on, and a six passenger vehicle. Furthermore, the analysis shows a possible short term benefit from larger capacity vehicles alleviating the airspace impact by reducing the number of simultaneous aircraft in the air. At the extreme the fleet size required to meet the same demand with a 10 passenger aircraft would be 61% smaller than a four passenger vehicle. This concept would align with NASA market studies of air-metro operations with prescribed departure times and routes (i.e., an airborne equivalent of a public bus service).

However, industry is expects a four passenger vehicle to be the most economical configuration and this report expands analysis on fleet sizes of 4,000, 5,000, and 6,000 aircraft. The expanded focus on understanding the utilization of the UAM vehicles as measured by the simultaneous operation in air, and the passenger delay experienced throughout the day. As seen in Figure 12 and 13 the 4,000 vehicle fleet operates at full utilization for the majority of the day, however it is not able to keep up with demand after 0800. These figures also show the 6,000 fleet size does not suffer from increasing delay to passenger on a per hour basis, as the fleet is able to keep up with demand and have a daily average below 15 minutes of delay to passengers.

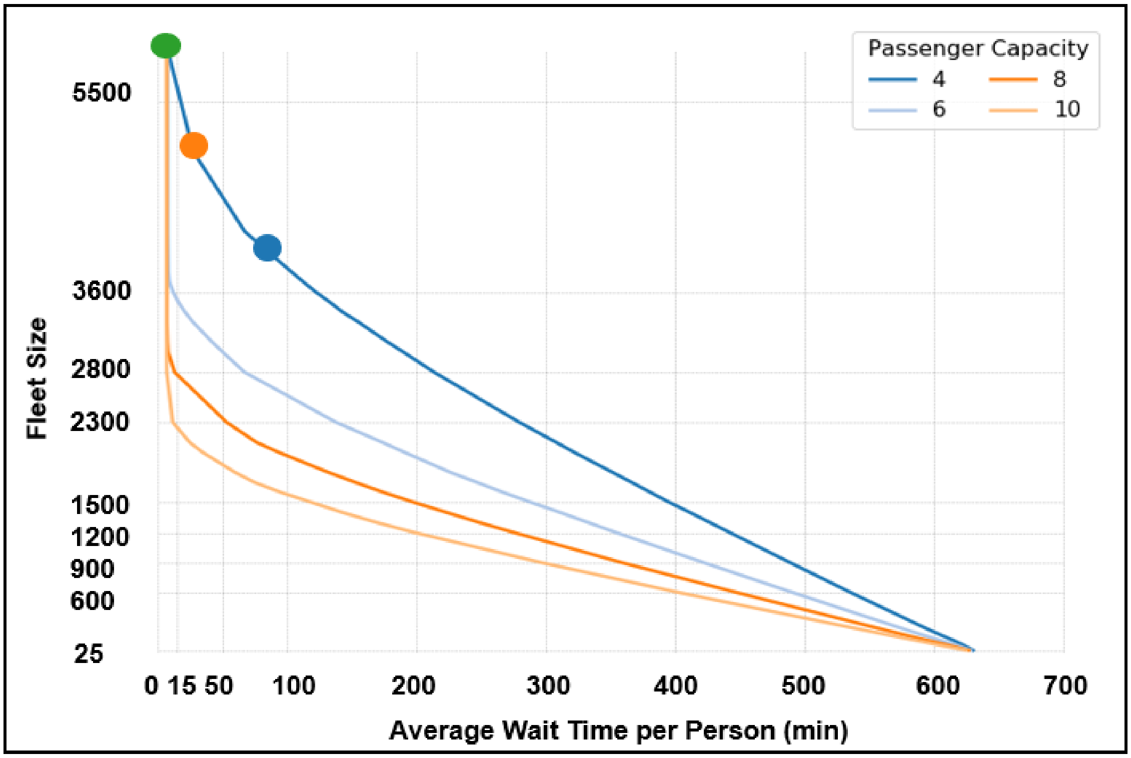

Figure 11. Evaluation of various fleet size and vehicle capacity versus the overall daily delay to passengers.

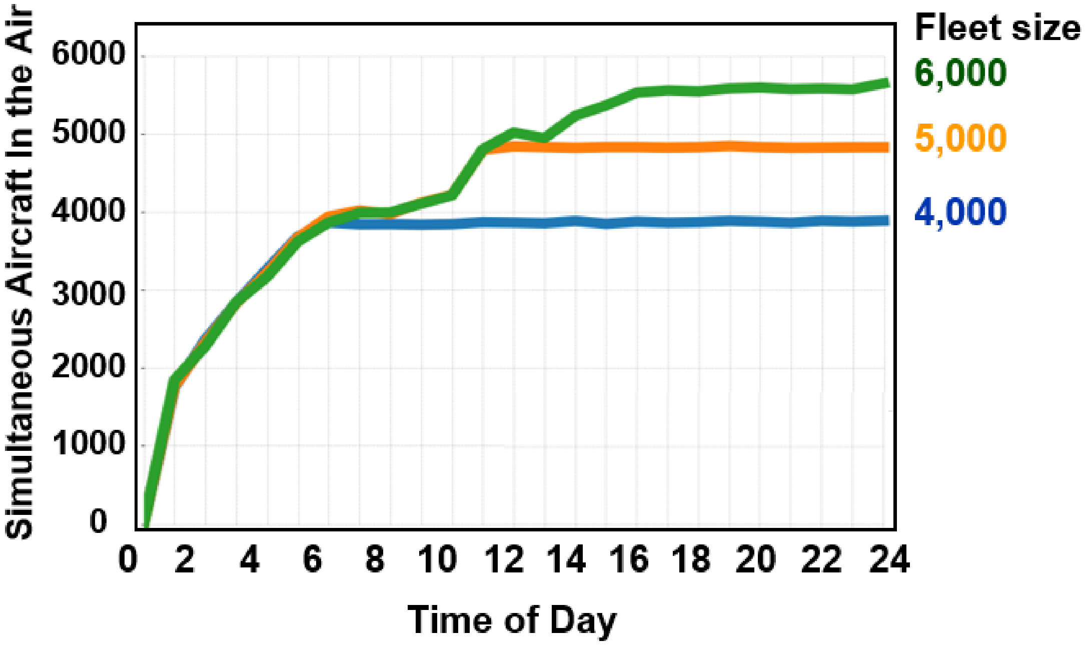

Figure 12. Simultaneous number of vehicles airborne at any given time of the day for a vehicle with a four passenger capacity.

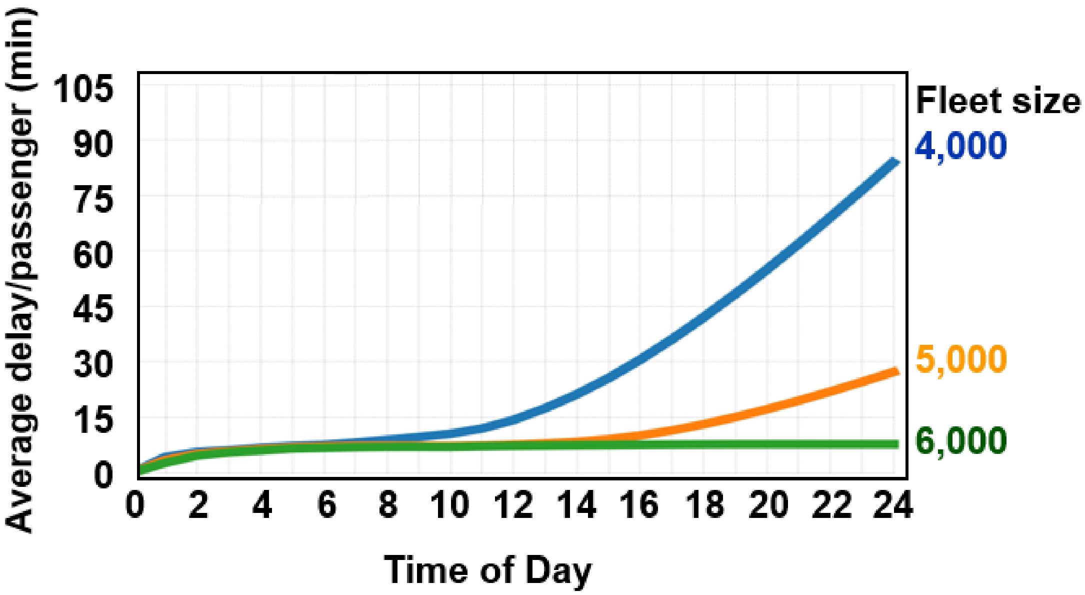

Figure 13. Average passenger delay at any given time of the day for a vehicle with a four passenger capacity.

This page intentionally left blank.

4. FUTURE WORK

In late 2020, RTCA special committee 228 (SC-228) created a ad-hoc working group to develop a concept of operation for UAM that the committee could build off to create standards for UAM operations. This CONOPS details more information about the corridor design for UAM [SC228CONOPS]. Specifically the CONOPS assumes corridors structures would be able to traverse altitudes from 1,000 - 5,000 ft. above ground level with structures as seen in Figure 14. As noted earlier, the corridor network at the moment assumes a single altitude of operation, and extension of this would consider corridors stacked vertically. In addition the cross-sections of such corridors could implement a similar overpass, underpass structure as seen in Figure 14. The concept of separation standards would also need to be introduced as it would reduce the throughput of aircraft in a corridor, and could thus leverage the vertical stacked corridors to release aircraft.

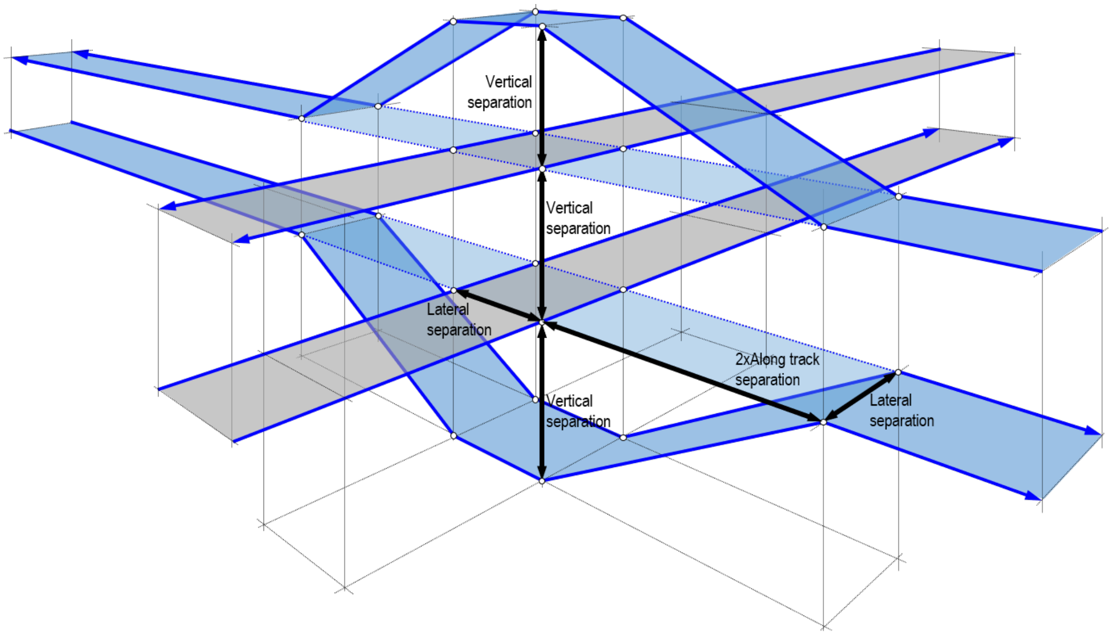

Figure 14. Excerpt from RTCA SC-228 UAM Concept of Operation V0.2. depicting possible corridor structures.

UAM operations are likely to suffer from low altitude weather events (i.e., convective weather, winds, low cloud cover) and should be considered when modeling traffic. For the upcoming year it is expected that wind information would be integrated to block sectors of a network that are unavailable due to high winds. This approach would use a cutoff criteria based on vehicle performance assumptions. Similar cutoff criteria could be developed for permeability and visibility.

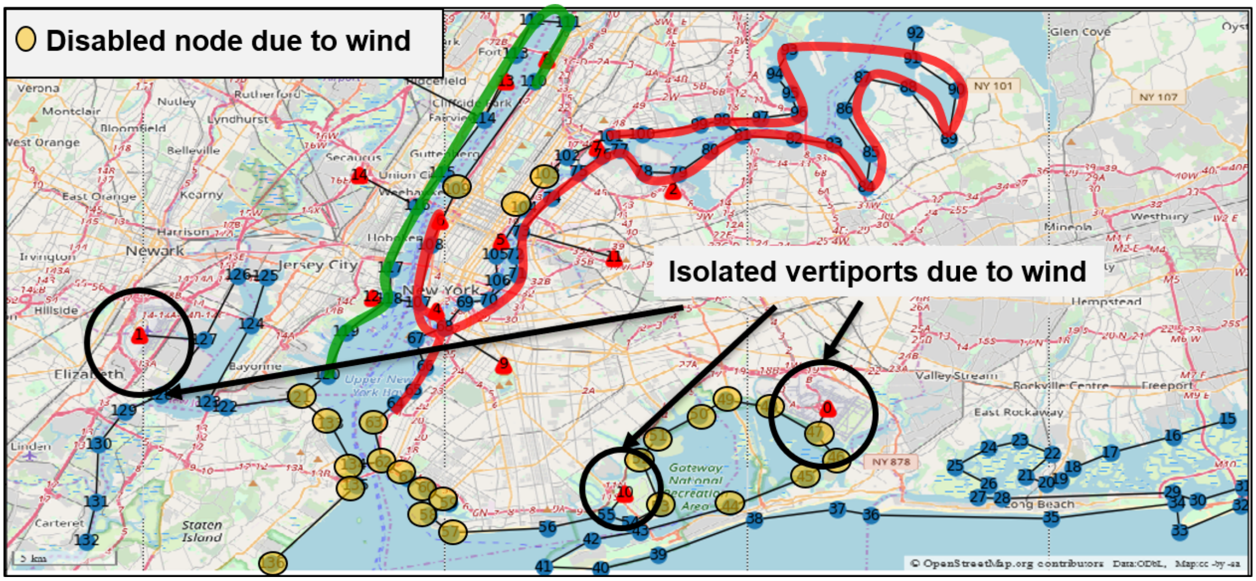
Disabled node due to wind
Isolated vertiports due to wind
© OpenStreetMap.org contributors

Figure 15. Demand ssignment.

In future years this tool is expected to include I) the ability to shut down sectors of a metropolitan region, II) a stack network of corridors to select a possible route, III) separation standards modeling to account for realistic throughput, IV) the ability to interface decision making tools that can deviate traffic outside of the defined corridors. Objective I is currently under development as part of FY21 funding and uses a G43 high resolution weather prediction model to provide wind information throughout the day. The wind information is used to shutdown each node in a network from 1,000 ft MSL to 5,000 MSL. This information will be used by our discrete event model to assign traffic to corridors that have zero shutdown risk during the scheduled flight window. Objective II is also under development in FY21 funding at is expected to reduce traffic throughput.

REFERENCES

[1] 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/.

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

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

[17] 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.

[18] 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.

[19] "New york city taxi and vehicle for hire database," *New York City Taxi Limousine Commission* (2019), URL https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page.

[20] T.W. Schneider, "Nyc taxi data," (2019), URL https://github.com/toddwschneider/nyc-taxi-data.git.

[21] L. Alvarez, "Uam tool kit github," (2019), URL git@llcad-github.llan.ll.mit.edu:lalvarez/UAMToolKit.git.

[22] Z. Zhou, D.S. Matteson, D.B. Woodard, S.G. Henderson, and A.C. Micheas, "A Spatio-Temporal Point Process Model for Ambulance Demand," *Journal of the American Statistical Association* 110(509), 6–15 (2015), URL https://ideas.repec.org/a/taf/jnlasa/v110y2015i509p6-15.html.

[23] N.R. Sturtevant and A. Felner, "A brief history and recent achievements in bidirectional search." (2018).