# Network of Multipurpose Docking Stations for Delivery Drones

António Miguel Teixeira da Conceição Caetano Dias
antonio.m.dias@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

May 2019

**Abstract**

Ever since Amazon manifested its desire to expand their delivery methods to drones, big strides towards developing the technology for modern society have been made. Drones are immune to road traffic and can operate autonomously which makes them very compelling for the e-commerce industry. However, delivery drones still face a few key obstacles. First, these are mostly being developed to deliver packages to customers living in properties with suitable, unobstructed areas where a drone can perform a delivery which immediately filters out potential customers living in properties ill-suited for it, such as apartment buildings. This is understood to be a very overlooked issue. Secondly, delivery drones are battery-powered which is a limiting factor when it comes to range, though it is in the interest of companies working on the technology to make it as accessible as possible. This work conceptualizes and implements an in-depth simulator of simplified networks of multipurpose docking stations for delivery drones that can both recharge their batteries (effectively extending their range, reaching more potential customers) and serve as pick-up points so that customers who do not live in properties suitable for conventional drone delivery can still benefit from the advantages of it as well as customers who simply find themselves in the proximity of a docking station and wish to have an item quickly delivered to them. The implementation developed returned data supporting the introduction of a network like this and measured its performance in several ways. Though there are limitations to what can be extracted from a simulation as opposed to real-world testing, such as the lack of accounting for some legal and urban limitations, it is believed that some of the results obtained present enough quality to act as general guidelines for future real-world implementations. Furthermore, the introduction of the pick-up point functionality proved to be extremely beneficial to the network by generally outperforming identically structured regular networks.
**Keywords:** network, delivery drones, docking stations, multipurpose, pick-up points.

## 1. Introduction

With the rapid development of UAV technology, several e-commerce companies have been making efforts to apply it to their businesses. The resulting benefits should translate into a positive balance when taking into account the initial cost of developing and implementing drones into this type of operations. [6] However, this technology is still at the early stages of development and there a few hurdles that need to be cleared in order to optimize its use. Obstacles like technical limitations of the hardware as well as barely present legislative frameworks pose a challenge for the celerity of its implementation. Sales made by e-commerce companies are expected to reach as much as USD 4 trillion by 2020, 15% of the total retail sales worldwide expected for the same year. [4] Nowadays these operations are generally optimized with efficient warehouse management. The most costly operation is still the last mile delivery, comprising on average 28% of the total cost of delivery. [4] The main idea is to reduce the costs of human labor by replacing conventional delivery systems of lightweight packages with delivery drones. These UAVs can also be optimized to deliver mail with improved accuracy and reduced human interference. [2] Battery technology is not expected to improve significantly past the general technological plateau that it currently sits at. [5] which is why it is crucial to find a sensible solution to fix the range problem that does not rely on a sudden leap in battery technology. Then there is the accessibility problem related to this technology. Delivery drones today are being developed to physically land and deliver packages to properties with suitable areas to accommodate for the operation, but a substantial portion of potential customers do not live in areas or residential buildings suitable for conventional drone delivery and negating a large number of potential customers the accessibility to the advantages that drone delivery has to offer is not in the interest of the companies that want to make it a reality. This study proposes the development of a network of multipurpose docking stations for delivery drones i.e. a network of physical stations where delivery drones can dock onto mid-flight to recharge their batteries, effectively extending their maximum ranges, however these docking stations can do more than just recharge as they are also capable of functioning as pick-up points where customers can go to retrieve their packages. This is a way to add versatility to the system as well as possible solution to customers who do not have conditions to benefit from conventional drone deliveries.

## 2. Delivery Drones

According to Amazon, its Prime Air development team has been testing a few different designs for their drones in different types of environments. [1] This suggests that Amazon Prime Air will have a fleet composed of more than one type of drone in order to increase the service's versatility. The depth of Amazon's fleet means that there is more than one set of technical specifications regarding its delivery drones. Unfortunately, Amazon does not provide easily accessible information on the technical aspects of their prototype drones and, apart from its first drone, the information available is limited. Its first prototype drone is known to have the technical specifications shown in Table 1 [3], [7].

A delivery drone inherently needs to be capable of highly precise VTOL operation to perform deliveries in modern urban areas. The need for VTOL is not only imposed by the operation itself, but also by the use of a docking station network. These stations need to cast a small footprint in order to be as

| | |
|---|---|
| **Propeller size (cm)** | 25.4 |
| **Propeller motors** | 8 |
| **Motor RPM** | 10000 |
| **Motor lift (N)** | approx. 15.69 |
| **Battery (estimation)** | 10000 mAh Li-Po |
| **Drone width (cm)** | 91.4 |
| **Package max weight (kg)** | 2.3 |
| **Max payload (kg)** | 14 |
| **Configuration** | multirotor |
| **Operating speed (km/h)** | 80 |
| **Flight altitude (m) (US)** | 122 m |
| **Flight time (min)** | 30 |
| **Range (km)** | 16 |

Table 1: Technical properties of Amazon Prime Air's first test drone. *Source: Jung et al., 2017 and Xu, 2017*

little intrusive as possible, mitigating any possibility of applying a sizable slingshot system or runway that fixed-wing drones usually demand. However, companies developing prototypes for delivery-drones are aware of the advantages that a fixed-wing configuration can offer over the relatively simple multirotor. As such, efforts are being made in order to come up with a configuration that can benefit from the controlled, precise delivery operation that a VTOL provides and the improved range, efficiency and speed of a fixed-wing design. As seen on Prime Air's promotional images, Amazon is already experimenting with what seems to be a hybrid configuration between a multirotor design and a fixed-wing design.

## 3. Implementation

### 3.1. The Drone

The delivery drone model developed for this work is defined by the properties shown in table 2.

| Designation | Symbol |
|---|---|
| Operating Speed [loaded] (km/h) | $U_{\text{loaded}}$ |
| Operating Speed [unloaded] (km/h) | $U_{\text{unloaded}}$ |
| Range [loaded] (km) | $R_{\text{loaded}}$ |
| Range [unloaded] (km) | $R_{\text{unloaded}}$ |
| Energy Consumption p/km [loaded] | $En_{\text{loaded}}$ |
| Energy Consumption p/km [unloaded] | $En_{\text{unloaded}}$ |
| Total Energy Consumption p/km | $En_{\text{total}}$ |
| Range Security Margin (%) | $RSM$ |
| Maximum Distance for Direct Delivery | $DD_{\max}$ |

Table 2: Delivery drone model.

The '[loaded]' designation represents a state where the drone is carrying a package. Consequently, the '[unloaded]' designation represents the opposite. The operating speeds $U_{\text{loaded}}$ and $U_{\text{unloaded}}$ are the speeds at which the drone moves while operating in each state. These values are set manually. As for range, two variables representing each flight state, $R_{\text{loaded}}$ and $R_{\text{unloaded}}$, are set. $R_{\text{loaded}}$ is set manually whilst $R_{\text{unloaded}}$ is calculated by manually indicating an improvement factor over the range in the '[loaded]' state.

$$R_{\text{loaded}} < R_{\text{unloaded}} \tag{1}$$

$$R_{\text{unloaded}} = R_{\text{loaded}} * \text{'improvement factor'} \tag{2}$$

The energy consumption represents a simplified way to gauge the amount of energy consumed by the drone per km of travel and can be calculated using (3). The total energy consumption $En_{\text{total}}$ is the sum of both (4) and represents a way to gauge the total round-trip consumption.

$$En_{[\text{state}]} = \frac{1}{R_{[\text{state}]}} \tag{3}$$

$$En_{\text{total}} = En_{\text{loaded}} + En_{\text{unloaded}} \tag{4}$$

The range security margin $RSM$ represents a safety cushion and is a simplistic way to account for a non-ideal performance return by the operation. It ensures that the simulations do not assume a perfect energy consumption rate since the drone is susceptible to a multitude of performance affecting factors in a real-world situation such as weather conditions, battery degradation, air traffic, etc. This value is set manually.

The maximum distance for direct delivery $DD_{\max}$ represents the maximum distance between the warehouse and the delivery address that allows for a direct delivery. This value is calculated using (5).

$$DD_{\max} = \frac{1 - RSM}{En_{\text{total}}} \tag{5}$$

This simulation is implemented in a way that allows for the alteration of the manually set values to gauge how these affect the returned results.

### 3.2. The Network

Using the delivery drone model developed, a strategy is defined in order to simulate a drone delivery network.

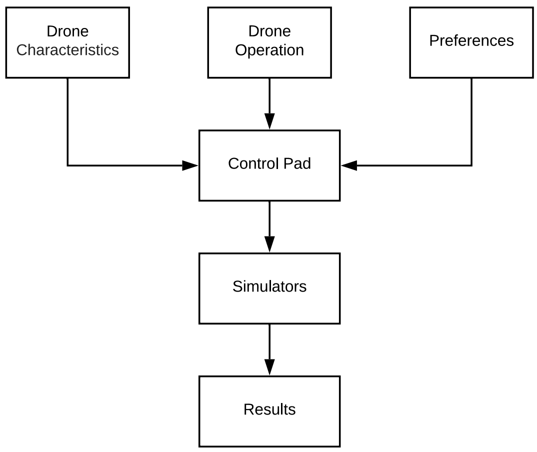


Figure 1: Implementation.

Every single simulation is dependant on a main 'Control Pad'. It contains the input values that the simulators fetch to generate and simulate the environments and return their results. It is composed of three main sections of values: the 'Drone Characteristics' section, the 'Drone Operation' section and the 'Preferences' section. Fig. 1 shows a graphical representation of the implementation. The 'Drone Characteristics' section corresponds to the delivery drone model. The 'Drone Operation' section contains the phases of operation performed by the drone along with the time period each phase takes to complete. These are set manually. It is defined in table 3.

$T_{\text{fcharge}}$ represents the time it takes to fully a charge a drone. $T_{\text{load}}$ corresponds to the time it takes to load the delivery drone with a package at the warehouse. $T_{\text{unload}}$ is the time it takes to safely perform the delivery of the package. $T_{\text{dock}}$ is the time it takes for the drone to dock onto either the warehouse once it completes its trip and returns, or a docking station. $T_{\text{undock}}$ is

| Drone Operation | |
|---|---|
| **Designation** | **Symbol** |
| Total Full Charge Time | $T_{\text{fcharge}}$ |
| Load Time | $T_{\text{load}}$ |
| Unload Time | $T_{\text{unload}}$ |
| Dock Time (Charge Only) | $T_{\text{dock}}$ |
| Undock Time (Charge Only) | $T_{\text{undock}}$ |
| Ascend Time | $T_{\text{ascend}}$ |
| Descend Time | $T_{\text{descend}}$ |

Table 3: 'Drone Operation' section.

the time it takes to perform the opposite action. $T_{\text{ascend}}$ is the time it takes for the drone to vertically ascend to its operation flight altitude whereas the $T_{\text{descend}}$ corresponds to the time it takes for the drone to descend from its operation flight altitude to dock onto a structure or perform a package delivery.

The 'Preferences' section is defined in table 4 and contains adjustable ratios to control the percentage of clients that place an order to be delivered to a docking station, and the demand area enlargement factor for pick-up point clients. These values are set manually.

| Preferences | |
|---|---|
| **Designation** | **Symbol** |
| % of Customers with Preference for Pick-up Point | $Cu_{\text{PuP}}$ |
| Enlarged Demand Area for Pick-up-Point Customers | $EA_{\text{PuP}}$ |

Table 4: 'Preferences' section.

$Cu_{\text{PuP}}$ is the percentage of the total demand that prefers to have their orders delivered to a docking station near them. These are customers that may live within range of eligibility for a conventional home delivery but prefer, for some reason, to retrieve their packages from the nearest docking station. This might be due to living in a residence ill-suited for drone delivery or because they simply need to get an item delivered to them quickly and it is convenient to pick it up at the closest docking station. If there is a docking station at a reasonable distance to a customer's residence, even if an order cannot be placed directly to that customer's location due to it being just outside the delivery area, he/she may still opt place an order to that docking station in order to benefit from the swiftness of drone delivery as opposed to regular delivery methods. These are customers that would not be eligible for drone delivery if the docking stations did not double as pick-up points since their homes/current locations are too far away to be eligible for a conventional drone delivery delivery. As such, an 'Enlarged Demand Area for Pick-up-Point Customers' $EA_{\text{PuP}}$ is created to extend the service's reach simply to account for this specific type of customer.

The simulator creates demand by randomly generating demand point coordinates within a coordinates range imposed by the range constraints inserted in the control pad. The implementation developed allows for the simulation of different environments and the area of operation is directly dependant on the structure of the network of stations. Adding one or more docking stations to the network considerably extends the area of operation because it allows the drone to fly farther, and depending on the strategy one decides to follow to geographically place the docking stations, the aspect of the area of operation can change drastically. Lastly, these docking stations can be given the ability to serve as pick-up points which also affects the area of operation and the way the demand behaves. Depending on the type of environment one wants to create, the simulator needs a specific set of data. If the environment simulated has at least one docking station in it, the simulator only needs to know the maximum distance a docking station can be from the warehouse ('Maximum Distance to Docking Station' or $DDS_{\max}$). This value is calculated using (6).

$$DDS_{\max} = \frac{1 - RSM}{En_{\text{loaded}}} \tag{6}$$

For any type of environment simulated, delivery drones are always fully recharged once they return to the warehouse. The reasoning behind this is that delivery drones are constantly being redeployed to perform more deliveries and recharging them after a completed mission ensures that every single drone is fully charged once it departs the warehouse. Table 5 shows which input values are needed to generate different types of environments.

| | Environment | | |
|---|---|---|---|
| **Values** | DD | DD + DS | DD + DS w/PuP |
| $DD_{\max}$ | Y | Y | Y |
| $DDS_{\max}$ | N | Y | Y |
| $Cu_{\text{PuP}}$ | N | N | Y |
| $EA_{\text{PuP}}$ | N | N | Y |

Table 5: Simulator input values used for different environments.

The 'DD' designation stands for direct delivery, 'DS' for docking station(s) and 'PuP' represents docking stations with a Pick-up Point functionality. A 'DD' environment is comprised of a warehouse and no docking stations. In this environment the only operation a delivery drone can perform is to directly deliver a package from the warehouse to the demand location, hence the 'DD' (direct delivery) designation. A 'DD + Docking Station(s)' environment is comprised of a warehouse and at least one docking station with no Pick-up Point functionality (only battery recharging). A 'DD + Docking Station(s) w/PuP' environment is comprised of a warehouse and at least one docking station. In this environment, all docking stations can also function as Pick-up Points.

Depending on the type of environment, a selection of the following drone delivery operations are possible:

**(i)**: **Direct Delivery** - Perform a direct delivery, if the location from where the order is placed is close enough to the warehouse do so.

**(ii)**: **Charging + Direct Return** - Perform a delivery where the drone has to dock onto a docking station to recharge its batteries before reaching the demand location to deliver its package, but is able to return directly to the warehouse after unloading the package.

**(iii)**: **Charging on Both Directions** - Perform a delivery where the drone has to recharge at a docking station both before delivering its package and after doing so.

**(iv)**: **Delivery at Docking Station** - Perform a delivery where the drone delivers the package directly to a docking station.

If an environment is such that only direct deliveries are possible, then this means that the delivery network does not contain any docking stations in it, but rather a warehouse, demand and delivery drones. This is why **(i)** is the only operation possible and the only data needed to generate this type of environment is $DD_{\max}$. This means that for a 'DD' type of environment, the coordinates range for the simulator is set as shown in table 6.

The warehouse is always set as the origin regardless of the type of environment.

| | Min | Max | Size |
|---|---|---|---|
| **X** | $X_{min} = -DD_{\max}$ | $X_{max} = +DD_{\max}$ | $S_X = -X_{min} + X_{max}$ |
| **Y** | $Y_{min} = -DD_{\max}$ | $Y_{max} = +DD_{\max}$ | $S_Y = -Y_{min} + Y_{max}$ |

Table 6: Coordinates range used by the simulator for a 'DD' environment.

$S_X$ and $S_Y$ are the effective width and length of the delivery area (Fig. 2).

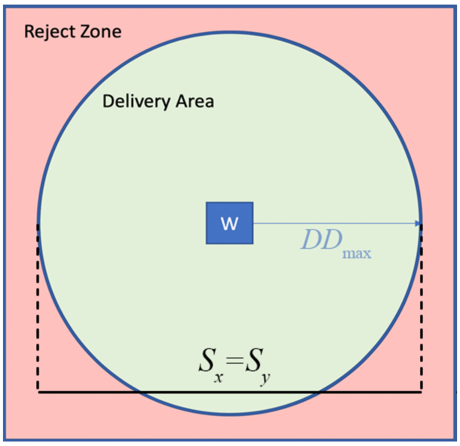


Figure 2: Delivery area for a 'DD' environment. The 'W' represents the warehouse.

The simulator then uses the information provided by the coordinates range to randomly generate demand locations. To do this, the *random()* function is used to generate values between 0 and 1, $De_{total}$ (total number of demand points/locations) times. Then, the coordinates that define each demand location, $X_{De}$ and $Y_{De}$, are generated as shown in table 7.

| Dem. Lo. | X | Y | X Dem. Coord. | Y Dem. Coord. |
|---|---|---|---|---|
| 1 | *rand()* | *rand()* | $X_{De_1} =$ $X_1 * S_X + X_{min}$ | $Y_{De_1} =$ $Y_1 * S_Y + Y_{min}$ |
| 2 | *rand()* | *rand()* | $X_{De_2} =$ $X_2 * S_X + X_{min}$ | $Y_{De_2} =$ $Y_2 * S_Y + Y_{min}$ |
| 3 | *rand()* | *rand()* | $X_{De_3} =$ $X_3 * S_X + X_{min}$ | $Y_{De_3} =$ $Y_3 * S_Y + Y_{min}$ |
| 4 | *rand()* | *rand()* | $X_{De_4} =$ $X_4 * S_X + X_{min}$ | $Y_{De_4} =$ $Y_4 * S_Y + Y_{min}$ |
| 5 | *rand()* | *rand()* | $X_{De_5} =$ $X_5 * S_X + X_{min}$ | $Y_{De_5} =$ $Y_5 * S_Y + Y_{min}$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $De_{total}$ | *rand()* | *rand()* | $X_{De_{[De_{total}]}} =$ $X_{[De_{total}]} * S_X + X_{min}$ | $Y_{De_{[De_{total}]}} =$ $Y_{[De_{total}]} * S_Y + Y_{min}$ |

Table 7: Generation of demand locations.

This results in $De_{total}$ scattered demand locations over a rectangular area. Since the delivery area is circular by nature, only some of these demand locations are eligible for drone delivery. To calculate the distance $D_{W-De}$ between a demand location and the warehouse, the Pythagorean theorem is applied (7).

$$D_{W-De} = \sqrt{X_{De}^2 + Y_{De}^2} \tag{7}$$

So, to determine if a demand location is within the valid delivery area for a direct delivery, condition (8) has to be true.

$$Cn_{DD} = D_{W-De} < DD_{\max} \tag{8}$$

If (8) is satisfied, the drone is capable to deliver to that demand location. If not, then that means the demand location is outside the delivery area and inside what Fig. 2 shows as the 'Reject Zone'. Fig. 3 is a timeline of a direct delivery operation, the only operation possible in a 'DD' environment.

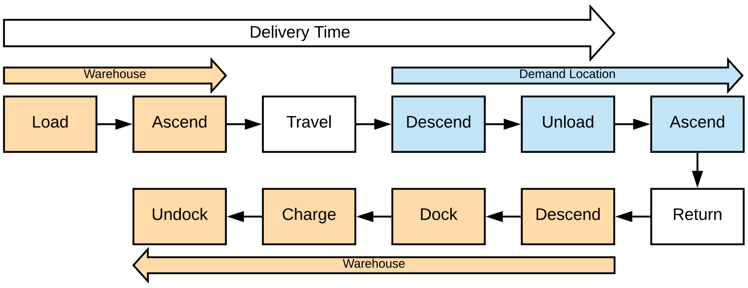


Figure 3: A direct delivery operation **(i)** which is the only type of operation possible in a 'DD' environment.

The operation time is counted until the drone fully charges after returning to the warehouse since only at that point it is ready to be used again. The travel time depends on the distance travelled by the drone, the speed at which it moves and the state the drone is in during that specific travel section (9).

$$T_{[\text{section}]} = \frac{'\text{distance travelled}'}{U_{[\text{state}]}} \tag{9}$$

Where '[section]' can be any section of the trip where the drone moves horizontally, such as moving from the warehouse to the demand point ($T_{\text{W}-\text{De}}$) or the return trip to the warehouse ($T_{\text{De}-\text{W}}$). In environments containing docking stations, time taken to move from and to stations, is calculated in the same way. The time it takes to recharge a drone back to full battery capacity once it docks onto the warehouse after having performed a direct delivery is calculated using (10).

$$T_{\text{charge}_{\text{DD}}} = [(2 * D_{W-De}) * En_{\text{total}}] * T_{\text{fcharge}} \tag{10}$$

The delivery time is the time it takes the drone to deliver the package i.e. the time passed between the departure from the warehouse until the instant the drone changes its state from '[loaded]' to '[unloaded]'. For a direct delivery, it is calculated using (11).

$$T_{\text{delivery}_{\text{DD}}} = T_{\text{load}} + T_{\text{ascend}} + T_{\text{W}-\text{De}} + T_{\text{descend}} + T_{\text{unload}} \tag{11}$$

The total operation time for a direct delivery $T_{\text{total}_{\text{DD}}}$ is the sum of the delivery time with the remaining operation time (12).

$$T_{\text{total}_{\text{DD}}} = T_{\text{delivery}_{\text{DD}}} + T_{\text{ascend}} + T_{\text{De}-\text{W}} + T_{\text{descend}} + \\ + T_{\text{dock}} + T_{\text{charge}_{\text{DD}}} + T_{\text{undock}} \tag{12}$$

As for the environment defined by 'DD + Docking Station(s)', this represents a region where delivery drones can dock onto docking stations to recharge (not deliver). Adding one or more docking stations to the delivery network enables operations **(i)**, **(ii)** and **(iii)**. To better understand how adding docking stations affects the way the simulation works, an environment where only one docking station exists is explained. Adding a docking station essentially means that the delivery area is extended. For

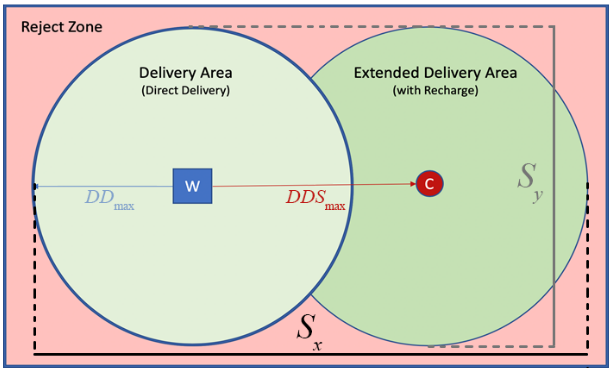


Figure 4: Delivery area for a 'DD + DS' environment with a single docking station. The 'W' represents the warehouse. The 'C' represents the docking station.

this case, a docking station is arbitrarily placed directly to the right of the warehouse (Fig. 4).

Since the docking station is directly to the right of the warehouse, the coordinates of both structures are as shown in table 8.

| | **X** | **Y** |
|---|---|---|
| **Warehouse** | 0 | 0 |
| **DS** | $X_{DS} = DDS_{\max}$ | $Y_{DS} = 0$ |

Table 8: Coordinates of the warehouse and the sole docking station placed to the right of the warehouse.

By adding a docking station to the environment, the coordinates range used by the simulator have to take $DDS_{\max}$ into account (table 9).

| | **Min** | **Max** | **Size** |
|---|---|---|---|
| **X** | $X_{min} =$ -$\mathrm{DD}_{\max}$ | $X_{max} =$ $+\mathrm{DD}_{\max} + DDS_{\max}$ | $S_X =$ -$X_{min} + X_{max}$ |
| **Y** | $Y_{min} =$ -$\mathrm{DD}_{\max}$ | $Y_{max} =$ $+\mathrm{DD}_{\max}$ | $S_Y =$ -$Y_{min} + Y_{max}$ |

Table 9: Coordinates range used by the simulator for a 'DD + Docking Station(s)' environment where the sole docking station is placed directly to the right of the warehouse.

For a direct delivery **(i)**, the docking station can be ignored by the drone since there is no need to recharge batteries mid-trip. The operation is the same as in the 'DD' environment (Fig. 3). As for operation **(ii)**, the drone needs to recharge its batteries before reaching the demand location but is able to return directly to the warehouse after doing so. This is due to a lesser stress on the motors when the drone is in its '[unloaded]' state (the energy consumption switches from $En_{\text{loaded}}$ to $En_{\text{unloaded}}$) since the payload becomes lighter after delivering the package, which means that the battery level the drone has after delivering the package is enough to fly back to the warehouse. However, even with a docking station, there is demand that can't be satisfied if their locations are outside the delivery area (seen as the 'Reject Zone' in Fig. 4) and these need to be filtered out. First, the distance between the demand location and the docking station needs to be calculated (13).

$$D_{De-DS} = \sqrt{[X_{De} - X_{DS}]^2 + [Y_{De} - Y_{DS}]^2} \tag{13}$$

In order to verify if the drone needs to dock onto a station before reaching its demand location, condition (14) needs to be true. It checks if a direct delivery is not possible (if (8) is false) and if the distance between the demand location and the docking station is less than $DD_{\max}$ so that the drone knows that it can reach the demand location after having replenished its batteries at the docking station.

$$Cn_{1DS} = \neg(Cn_{DD}) \wedge (D_{De-DS} < DD_{\max}) \tag{14}$$

Now, to verify if the drone is able to directly return to the warehouse after delivering the package (i.e. perform operation **(ii)**), condition (15) needs to be true. It checks if the drop in energy consumption rate after the drone delivers its package is significant enough that the drone is able to return directly to the warehouse without docking onto a station again. This condition is only relevant if the drone had to also dock before reaching the demand location (or else a direct delivery would have been possible all along) which is why (14) has to be true.

$$Cn_{DR} = [(D_{De-DS} * En_{\text{loaded}} + D_{W-De} * En_{\text{unloaded}}) < (1 - RSM)] \wedge Cn_{1DS} \tag{15}$$

So, it is now easy to make a condition that verifies if a drone needs to recharge its batteries again after delivering its package i.e. (16) has to be true. It checks if the drone recharged its batteries before delivery and if it cannot return directly to the warehouse after delivery.

$$Cn_{2DS} = Cn_{1DS} \wedge \neg Cn_{DR} \tag{16}$$

Table 10 shows a logic board explaining how the conditions defined above control which kind of operation the drone performs given demand location or if it can't perform any operation at all.

| | $Cn_{DD}$ | $Cn_{1DS}$ | $Cn_{DR}$ | $Cn_{2DS}$ |
|---|---|---|---|---|
| **(i)** | T | F | F | F |
| **(ii)** | F | T | T | F |
| **(iii)** | F | T | F | T |
| **Reject Zone** | F | F | F | F |

Table 10: Logic table with conditions (8), (14), (15) and (16), and the values they return for each kind of operation.

Regarding operation times, for operation **(i)**, these are calculated in the same way as in the 'DD' environment using (11) for the delivery time $T_{\text{delivery}_{\text{DD}}}$ and (12) for the total operation time $T_{\text{total}_{\text{DD}}}$.

$$T_{\text{delivery}_{(\text{i})}} = T_{\text{delivery}_{\text{DD}}} \tag{17}$$

$$T_{\text{total}_{(\text{i})}} = T_{\text{total}_{\text{DD}}} \tag{18}$$

Fig. 5 shows operation **(ii)**.

The delivery time $T_{\text{delivery}_{(\text{ii})}}$ has to account for the stop at the docking station. To do this, the time it takes to recharge the drone at a docking station before it delivers its package (while it still is in its '[loaded]' state) needs to be calculated using (19).

$$T_{\text{charge}_{1\text{DS}}} = (DDS_{\max} * En_{\text{loaded}}) * T_{\text{fcharge}} \tag{19}$$

The time it takes to recharge the batteries at the warehouse after directly returning from the demand location is calculated using (20).

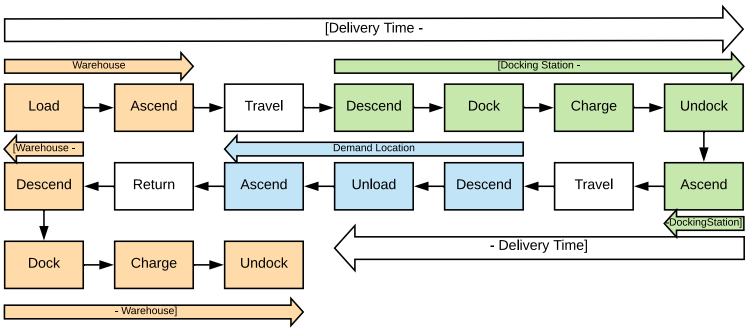


Figure 5: Delivery operation **(ii)** where the drone needs to recharge its batteries before reaching the demand location.

$$T_{\text{charge}_{\text{DR}}} = [D_{DS-De} * En_{\text{loaded}} + D_{W-De} * En_{\text{unloaded}}] * T_{\text{fcharge}} \quad (20)$$

The delivery time for operation **(ii)** is then calculated using (21).

$$T_{\text{delivery}_{\text{(ii)}}} = T_{\text{load}} + T_{\text{ascend}} + T_{\text{W-DS}} + T_{\text{descend}} + T_{\text{dock}} + T_{\text{charge}_{\text{1DS}}} + T_{\text{undock}} + T_{\text{ascend}} + T_{\text{DS-De}} + T_{\text{descend}} + T_{\text{unload}} \quad (21)$$

The total operation time $T_{\text{total}_{\text{(ii)}}}$ for this operation is obtained by adding the remaining operation steps (22).

$$T_{\text{total}_{\text{(ii)}}} = T_{\text{delivery}_{\text{(ii)}}} + T_{\text{ascend}} + T_{\text{De-W}} + T_{\text{descend}} + T_{\text{dock}} + T_{\text{charge}_{\text{DR}}} + T_{\text{undock}} \quad (22)$$

In operation **(iii)** the drone docks onto a station twice ((14) and (16) are true). This changes the way it operates yet again (Fig. 6).

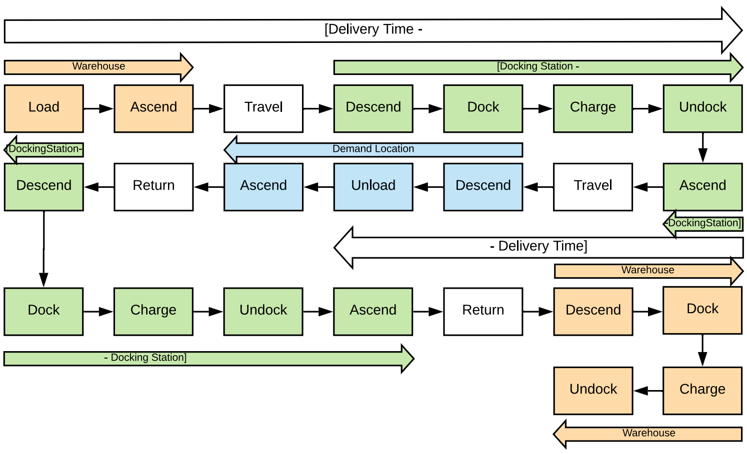


Figure 6: Delivery operation **(iii)** during which the drone needs to recharge its batteries before and after reaching the demand location.

The delivery time for operation **(iii)** $T_{\text{delivery}_{\text{(iii)}}}$ is calculated in the exact same way as for operation **(ii)**, since the operation until the delivery is exactly the same (23).

$$T_{\text{delivery}_{\text{(iii)}}} = T_{\text{delivery}_{\text{(ii)}}} \quad (23)$$

However, the total operation time $T_{\text{total}_{\text{(iii)}}}$ now has to account for the second stop at the docking station. As such, the time it takes to charge the drone at the docking station after delivering the package and before returning to the warehouse needs to be calculated using (24).

$$T_{\text{charge}_{\text{2DS}}} = [D_{DS-De} * En_{\text{loaded}} + D_{DS-De} * En_{\text{unloaded}}] * T_{\text{fcharge}} \quad (24)$$

So, the time it takes to charge the delivery drone once it returns to the warehouse for operation **(iii)** is simply calculated using (25).

$$T_{\text{charge}_{\text{2DS}_\text{W}}} = DDS_{\text{max}} * En_{\text{unloaded}} * T_{\text{fcharge}} \quad (25)$$

The total operation time for operation **(iii)** $T_{\text{total}_{\text{(iii)}}}$ is obtained by adding the remaining operation steps to the $T_{\text{delivery}_{\text{(iii)}}}$ (26).

$$T_{\text{total}_{\text{(iii)}}} = T_{\text{delivery}_{\text{(iii)}}} + T_{\text{ascend}} + T_{\text{De-DS}} + T_{\text{descend}} + T_{\text{dock}} + T_{\text{charge}_{\text{2DS}}} + T_{\text{undock}} + T_{\text{ascend}} + T_{\text{DS-W}} + T_{\text{descend}} + T_{\text{dock}} + T_{\text{charge}_{\text{2DS}_\text{W}}} + T_{\text{undock}} \quad (26)$$

Adding multiple docking stations to the environment allows for the same operations as in the case with one single docking station, with the difference being that the simulator now needs to select which docking station to use for each demand location. In order to demonstrate how the system handles multiple docking stations, a hexagonal grid of six docking stations is defined in Fig. 7. The simulator can handle any placement of docking stations, but a hexagonal grid distribution is particularly adequate for a network of this kind. The coordinates of each docking station in the network are defined in table 11 along with the warehouse (origin). The coordinates range used by this network is shown by table 12.

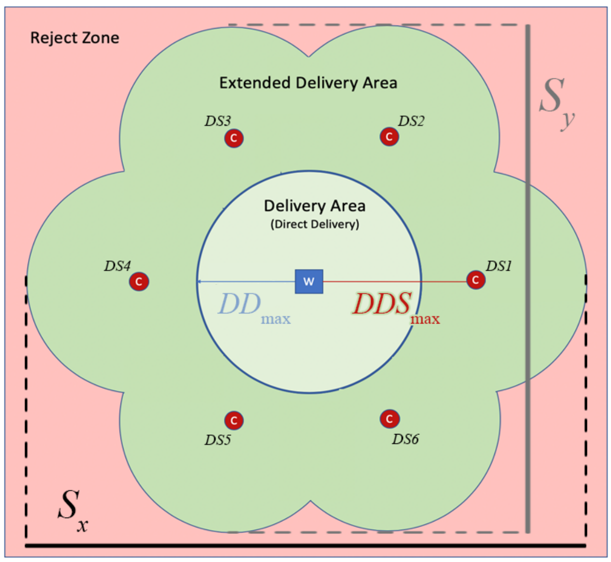


Figure 7: Delivery area for a 'DD + DS' environment with a hexagonal grid of six docking stations. The 'W' represents the warehouse. The 'C' represents a docking station.

The same exact operations (**(i)**, **(ii)** and **(iii)**) are possible in this environment as in the one with a single docking station, with

| | X | Y |
|---|---|---|
| **Warehouse** | 0 | 0 |
| **DS1** | $DDS_{\max}$ | 0 |
| **DS2** | $DDS_{\max}/2$ | $\sqrt{DDS_{\max}^2 + (DDS_{\max}/2)^2}$ |
| **DS3** | $-DDS_{\max}/2$ | $\sqrt{DDS_{\max}^2 + (DDS_{\max}/2)^2}$ |
| **DS4** | $-DDS_{\max}$ | 0 |
| **DS5** | $-DDS_{\max}/2$ | $-\sqrt{DDS_{\max}^2 + (DDS_{\max}/2)^2}$ |
| **DS6** | $DDS_{\max}/2$ | $-\sqrt{DDS_{\max}^2 + (DDS_{\max}/2)^2}$ |

Table 11: Coordinates of the warehouse and the six docking stations placed on a hexagonal grid around it.

the only difference being that there are several docking stations available, extending the delivery area even more. This means that, for operations **(ii)** and **(iii)**, there needs to be some kind of way to determine which docking station is used for each demand point location. This is very simply done by calculating the distance from each demand point to every single docking station available using (13).

| | Min | Max | Size |
|---|---|---|---|
| **X** | $X_{min} = -(DD_{\max} + DDS_{\max})$ | $X_{max} = +DD_{\max} + DDS_{\max}$ | $S_X = -X_{min} + X_{max}$ |
| **Y** | $Y_{min} = -(\sqrt{DDS_{\max}^2 + (DDS_{\max}/2)^2}$ +DDmax) | $Y_{max} = \sqrt{DDS_{\max}^2 + (DDS_{\max}/2)^2}$ +DDmax | $S_Y = -Y_{min} + Y_{max}$ |

Table 12: Coordinates range used by the simulator for a 'DD + DS' environment with six docking stations placed on a hexagonal grid around the warehouse.

Once all the six distance values are calculated, a *min()* function is applied to determine which is the smallest i.e. which docking station is closest to the demand location. The closest docking station is the one that ends up being selected for that specific demand location. For an environment with six docking stations, the function is run as (27).

$$min(D_{De-DS1}, D_{De-DS2}, D_{De-DS3}, D_{De-DS4}, D_{De-DS5}, D_{De-DS6}) \quad (27)$$

After determining which docking station is the closest, the simulator tests conditions (8), (14), (15) and (16) to determine which kind of operation the drone needs to perform according to table 10 or if can't perform any operation at all. So, for each demand location in the environment, the distance to the warehouse and the distances to every single docking station in the environment are simultaneously calculated. Then, the simulator checks if a direct delivery **(i)** is possible. If not, the simulator uses the conditions previously mentioned to check which type of operation (**(ii)** or **(iii)**) can be performed by using the closest docking station to the demand location. If no operation is possible, this means that the demand location is outside the delivery area and all conditions returned false i.e. it is inside the 'Reject Zone' depicted in Fig. 7. The delivery time and total operation time for this environment are calculated in the exact same way as in the environment with a single docking station since after the docking station selection process is done, the operations are identical. This means that for operation **(i)** in an environment with multiple docking stations, the drone perceives it as the environment in Fig. 3 since the system ignores the docking stations. The delivery time and total operation time are calculated using (11) and (12) respectively. Operation **(ii)** is shown in Fig. 5 and its delivery time and total operation time are calculated using (21) and (22) respectively. Similarly, operation **(iii)** is shown in Fig. 6 and its delivery time is calculated using (23) and the total operation time is returned by (26).

Having the docking stations function as pick-up points, slightly changes the way the network works. The fact that orders can be delivered directly to a docking station means that, even if the delivery area remains the same, some customers living in the 'Reject Zone' may willingly choose to physically go inside the delivery area to pick-up their orders at one of the available docking stations. Furthermore, customers who live inside the delivery area may also use the docking stations as pick-up points whether because they do not live in a residence suited for a conventional drone delivery or because they are located somewhere else and want to get an order quickly, for which the only short-term solution is drone delivery.

Making use of the environment previously shown in Fig. 4, the docking stations are upgraded to also act as pick-up points (Fig. 8). As mentioned before, the simulator needs the two extra input values from the 'Preferences' section of the 'Control Pad' to simulate this environment, the '% of Customers with Preference for Pick-up Point' $Cu_{\text{PuP}}$ and the 'Enlarged Demand Area for Pick-up-Point Customers' $EA_{\text{PuP}}$.

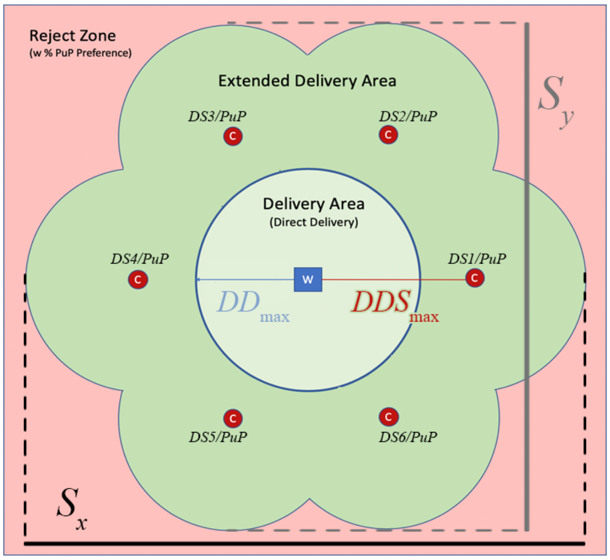


Figure 8: Delivery area for a 'DD + DS w/PuP' environment with a hexagonal grid of six docking stations capable of functioning as pick-up points. The 'W' represents the warehouse. The 'C' represents a docking station.

The coordinates of the warehouse and the six docking stations are obviously the same as in the previous environment. These are shown in table 11. The coordinates range for the simulator needs to be adapted for the 'Enlarged Demand Area for Pick-up-Point Customers' as shown in table 13. This is simply done by multiplying the axis limits by $\sqrt{1 + EA_{\text{PuP}}}$ where $1 + EA_{\text{PuP}}$ factors in the enlargement of the considered area and the square root is then used to extract a length increase factor from an area increase factor.

In this environment, every single type of operation possible in the 'DD' (**(i)**) and DD + Docking Station(s) (**(i)**, **(ii)** and **(iii)**) environments is possible and done in the exact same way. However, this environment introduces the possibility of delivery to a docking station, which corresponds to operation **(iv)**. In order to adapt to it, another condition (28) needs to be tested by

| | Min | Max | Size |
|---|---|---|---|
| **X** | $X_{min} = [-(DD_{\max} + DDS_{\max})] \sqrt{1 + EA_{\text{PuP}}}$ | $X_{max} = (+DD_{\max} + DDS_{\max}) \sqrt{1 + EA_{\text{PuP}}}$ | $S_X = -X_{min} + X_{max}$ |
| **Y** | $Y_{min} = [-(\sqrt{DDS^2_{\max} + (DDS_{\max}/2)^2} + \text{DDmax})]{*}\sqrt{1 + EA_{\text{PuP}}}$ | $Y_{max} = [\sqrt{DDS^2_{\max} + (DDS_{\max}/2)^2} + \text{DDmax}]{*}\sqrt{1 + EA_{\text{PuP}}}$ | $S_Y = -Y_{min} + Y_{max}$ |

Table 13: Coordinates range used by the simulator for a 'DD + Docking Station(s) w/PuP' environment with six docking stations placed on a hexagonal grid around the warehouse.

the simulator. This condition determines if the drone performs a delivery to a docking station or not.

$$Cn_{PuP} = [random() < Cu_{PuP}] \wedge \neg Cn_{DD} \tag{28}$$

It filters customers that can't order via direct delivery and are within the '% of Customers with Preference for Pick-up Point'. The first term randomly determines if the customer in that demand location is within the group of customers that have a preference for retrieving orders from a docking station whilst the second term tests if direct delivery is not possible. Table 10 is updated to include the new condition, resulting in table 14.

| | $Cn_{DD}$ | $Cn_{1DS}$ | $Cn_{DR}$ | $Cn_{2DS}$ | $Cn_{PuP}$ |
|---|---|---|---|---|---|
| **(i)** | T | F | F | F | F |
| **(ii)** | F | T | T | F | F |
| **(iii)** | F | T | F | T | F |
| **(iv)** | F | F | F | F | T |
| **Reject Zone** | F | F | F | F | F |

Table 14: Logic table with conditions (8), (14), (15), (16) and (28), and the values they return for each kind of operation.

The delivery and operation times for operation **(iv)** are obtained in a slightly different way. To better understand why, Fig 9 shows the steps and timeline for this kind of operation.

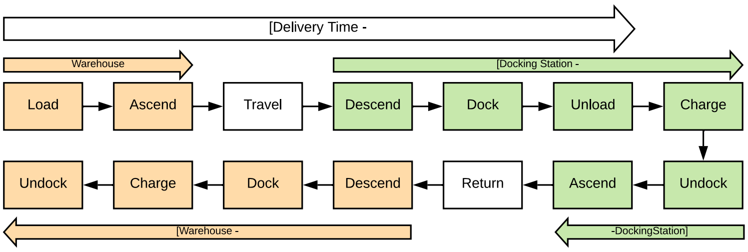


Figure 9: Delivery operation **(iv)** during which the drone delivers its package to a docking station.

The delivery time $T_{\text{delivery}_{(\text{iv})}}$ is now counted until the point when the delivery drone delivers its package to the docking station.

$$T_{\text{delivery}_{(\text{iv})}} = T_{\text{load}} + T_{\text{ascend}} + T_{\text{W-DS}} + T_{\text{descend}} + T_{\text{dock}} + T_{\text{unload}} \tag{29}$$

Finally, the total operation time for this operation is calculated using (30).

$$T_{\text{total}_{(\text{iv})}} = T_{\text{delivery}_{(\text{iv})}} + T_{\text{charge}_{1\text{DS}}} + T_{\text{undock}} + T_{\text{ascend}}$$
$$+ T_{\text{DS-W}} + T_{\text{descend}} + T_{\text{dock}} + T_{\text{charge}_{2\text{DS}_\text{W}}} + T_{\text{undock}} \tag{30}$$

## 4. Results

In this section, results returned by the implementation are analyzed and discussed. In order to cover all the different environments and present them in this format, results are based on the average of values returned by the simulations for 25000 randomly generated demand point locations for each of the four environments described in section 3.1 (for a total of 100000 randomly generated demand points). For simplicity, in this first base run the operating speed in the unloaded state is considered to be the same ($U_{\text{unloaded}} = U_{\text{loaded}} = 80$ km/h). The range improvement factor is set at 20% ($R_{\text{loaded}} = 16$ km, $R_{\text{unloaded}} = R_{\text{loaded}} * 1.2 = 19.2$ km). The range security margin $RSM$ is set at 15% ($RSM = 15\%$). The remaining 'Control Pad' values are shown in table 15. Drone Operation times were set arbitrarily to plausible values. $T_{\text{load}} > T_{\text{unload}}$ since, based on Amazon's promotional videos, drones are loaded manually by a human operator. A fast charging system is assumed accounting for the technological developments since Amazon first announced their Prime Air division and, as such, $T_{\text{fcharge}}$ is set at an optimistic 20 minutes. The docking mechanism is set to take 30 seconds to operate both ways $T_{\text{undock}} = T_{\text{dock}}$. Both $T_{\text{ascend}}$ and $T_{\text{descend}}$ are set at 10 seconds. Using the input values shown in Fig. 15, the 'Operation Map' returned for each environment is shown in table 18.

| **Drone Characteristics** | |
|---|---|
| $U_{\text{loaded}}$ | 80 km/h |
| $U_{\text{unloaded}}$ | 80 km/h |
| $R_{\text{loaded}}$ | 16 km |
| $R_{\text{unloaded}}$ (IF = 1.2) | 19.2 km |
| $En_{\text{loaded}}$ | 6.3% |
| $En_{\text{unloaded}}$ | 5.2% |
| $En_{\text{total}}$ | 11.5% |
| $RSM$ | 15% |
| $DD_{\max}$ | 7.4 km |
| **Drone Operation** | |
| $T_{\text{fcharge}}$ | 20 min |
| $T_{\text{load}}$ | 1 min. |
| $T_{\text{unload}}$ | 30 secs. |
| $T_{\text{dock}}$ | 30 secs. |
| $T_{\text{undock}}$ | 30 secs. |
| $T_{\text{ascend}}$ | 10 secs. |
| $T_{\text{descend}}$ | 10 secs. |
| **Preferences** | |
| $Cu_{\text{PuP}}$ | 30% |
| $EA_{\text{PuP}}$ | 20% |

Table 15: Control Pad input values for a drone model inspired on one of Amazons Prime Air drones.

A summary of the results returned by the simulator is shown in Tables 16 and 17.

Fig 10 shows a bitmap image of the '1W+6DS w/PuP' environment with the delivery area and the 'Reject Zone' overlaid on top of it.

Analyzing the results for the set of input values considered in the 'Control Pad', it is possible to see a an increase in the average operation time for first three environments. This is caused by the considerable increase in delivery area due to network growth. The delivery drones are able to fly farther the larger the delivery area, but they move at the same speeds in all the environments which is why the operation times are generally higher on average each time the delivery area increases. However, the fourth environment is structurally the same as the third with the added fea-

| Environment | Coverage (km$^2$) | Rel. Demand | Avg. Op. Time | Avg. Delivery Time | Avg. Airborne Time | Deliveries/Hour/Drone |
|---|---|---|---|---|---|---|
| 1W | 172.9 | 100 | 0:21:52 | 0:05:32 | 0:08:04 | 2.74 |
| 1W+1DS | 340.8 | 197 | 0:46:55 | 0:19:36 | 0:17:47 | 1.28 |
| 1W+6DS | 1160.9 | 671 | 1:04:44 | 0:29:46 | 0:24:42 | 0.93 |
| 1W+6DS w/PuP | 1393.0 | 806 | 0:59:47 | 0:22:43 | 0:17:41 | 1.00 |

Table 16: A summary of the simulations for each environment.

| | Environment | | | |
|---|---|---|---|---|
| Mission | 1W | 1W+1DS | 1W+6DS | 1W+6DS w/PuP |
| (i) | 100% | 51% | 15% | 13% |
| (ii) | - | 12% | 22% | 15% |
| (iii) | - | 37% | 63% | 39% |
| (iv) | - | - | - | 33% |

Table 17: Distribution of type of operation per environment.

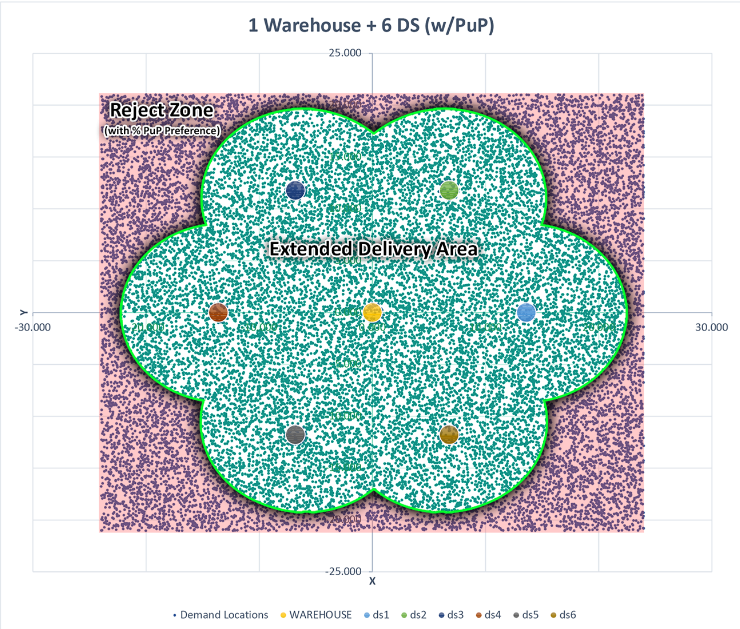


Figure 10: Scatter plot of the '1W+6DS (w/PuP)' environment showing the 25000 individual demand points, the six docking stations and the warehouse.

| | 1W | 1W+1DS | 1W+6DS | 1W+6DS w/PuP |
|---|---|---|---|---|
| $S_X$ | 14.8 | 28.4 | 42.0 | 47.9 |
| $S_Y$ | 14.8 | 14.8 | 38.4 | 42.1 |
| Coordinates | | | | |
| **W** | (0;0) | | | |
| **DS1** | - | (13.6;0) | (13.6;0) | |
| **DS2** | - | - | (6.8;11.8) | |
| **DS3** | - | - | (-6.8;11.8) | |
| **DS4** | - | - | (-13.6;0) | |
| **DS5** | - | - | (-6.8;-11.8) | |
| **DS6** | - | - | (6.8;-11.8) | |

Table 18: 'Operation Map' for each of the four implemented environments using the input values from Table 15.

ture of having PuP-capable docking stations though there is still a decrease in both the average operation time and the average delivery time. The average delivery time in every environment is consistently lower than 30 minutes, which satisfies Amazon's desire to deliver orders via drone in less than 30 minutes, and taking into account that the real life based values used in the 'Control Pad' are most likely from Amazon's first prototype test-ready drone, these results are very positive.

Figures 12, 13, 14 and 15 show data returned by testing ranges of different 'Control Pad' input values.

## 5. Conclusions

Overall, in the context of the implementation developed, the results obtained show a definite functional benefit in using a network of docking stations for delivery drones as well as the inclusion of a pick-up point functionality to it. In the first base analysis, input data inspired on Amazon's Prime Air first prototype was applied to the drone model developed, and tested for the four environments described in section 3.1. The results show a perfectly plausible network in the sense that they provide a good justification for drone delivery, even when considering the inherent shortcomings of a simulator when compared to a real-world application.

Although the implementation and results explained in this work enforce it, this document is not to be seen as another study that backs the introduction of drone delivery to modern society, but rather as a work that assumes its imminent introduction and helps gauge how a network like this may scale and behave. Furthermore, it intends to improve upon it by conceptualizing and implementing a new crucial functionality (PuP-enabled docking stations).

First, it is important to be aware of the limitations carried by the implementation developed when compared to a real-world application. There are several aspects that are not taken into account by it. In a real-world situation, drones can't always fly in straight lines, they need to be aware of their peers as well as flight restricted zones and obstacles. There will most likely exist flight corridors reserved for drone-based commercial operations which should incur in a delivery area decrease depending on how these corridors are defined since drones won't always be able to go from point A to point B in straight lines, but rather in segments of flight corridors. Properties like terrain irregularity may demand continuous altitude adjustments for delivery drones in order to minimize noise and ensure a secure operation. Weather conditions can have a very strong impact on a network of delivery drones and may even demand for a complete operation halt in severe cases. The time it takes to perform certain operation steps may be very different and susceptible to variation. However, given the scope of this study's objective and the inherent limitations of this work format, it is understood by the author that the implementation developed was able to provide very useful data, some of which with enough quality to act as a general guideline when idealizing the type of structure discussed in this work.

Apart from the complete development of the implementation

**Fleet Dimensions**

| 1 Warehouse (DD) | | | |
|---|---|---|---|
| Requests/Hour | Minimum Number of Active Drones | Sky Congestion ('Flying Drones'/Km^2) | Warehouse Congestion |
| 1 | 1 | 0.00 | 0.23 |
| 10 | 4 | 0.01 | 2.27 |
| 100 | 37 | 0.08 | 22.73 |
| 1000 | 365 | 0.78 | 227.25 |
| 10000 | 3645 | 7.77 | 2,272.53 |

| 1 Warehouse + 6 DS | | | |
|---|---|---|---|
| Requests/Hour | Minimum Number of Active Drones | Sky Congestion ('Flying Drones'/Km^2) | Warehouse Congestion |
| 1 | 2 | 0.000 | 0.27 |
| 10 | 11 | 0.004 | 2.72 |
| 100 | 108 | 0.035 | 27.23 |
| 1000 | 1079 | 0.355 | 272.26 |
| 10000 | 10789 | 3.547 | 2,722.60 |

| 1 Warehouse + 1 DS | | | |
|---|---|---|---|
| Requests/Hour | Minimum Number of Active Drones | Sky Congestion ('Flying Drones'/Km^2) | Warehouse Congestion |
| 1 | 1 | 0.00 | 0.23 |
| 10 | 8 | 0.009 | 2.54 |
| 100 | 79 | 0.087 | 25.39 |
| 1000 | 782 | 0.870 | 253.87 |
| 10000 | 7819 | 8.698 | 2,538.73 |

| 1 Warehouse + 6 DS (w/PuP) | | | |
|---|---|---|---|
| Requests/Hour | Minimum Number of Active Drones | Sky Congestion ('Flying Drones'/Km^2) | Warehouse Congestion |
| 1 | 1 | 0.000 | 0.27 |
| 10 | 10 | 0.002 | 2.71 |
| 100 | 100 | 0.021 | 27.14 |
| 1000 | 996 | 0.212 | 271.41 |
| 10000 | 9958 | 2.117 | 2,714.11 |

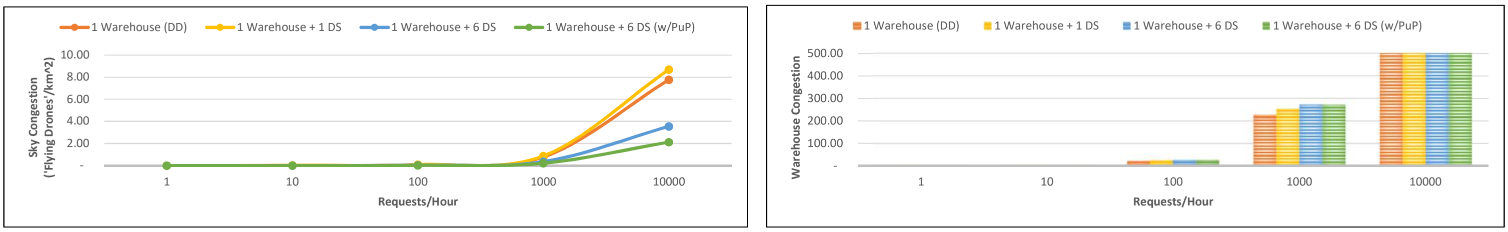


Figure 11: Fleet dimensions behaviour.

itself, this work's introduction of deliveries/hour/drone as a qualitative indicator may be almost as important as some of the results and can represent a very powerful tool to dimensionalize networks of this sort in the future.

For the base analysis, the average delivery times for the four environments satisfy Amazon's desired delivery time of 30 minutes or less since every single environment returned an average delivery time below 30 minutes. The inclusion of docking stations enables multiple types of operation since drones have physical structures at their disposal onto which they can dock to recharge batteries. Operations like 'Charging + Direct Return' and 'Charging on Both Directions' in which the drone recharges either strictly before or both before and after delivering its package show the new-found versatility that docking stations provide beyond acting simply as a 'fuel' replenishment station. These new operations enabled by the stations equate to more reach, drones can go much farther which in turn makes drone delivery more accessible to more potential customers. Drone technology cannot rely on a sudden leap in battery technology, workarounds like docking stations are a short-term solution and may actually be a long-term one due to the limitations of current and near-future battery technology. The very steep growth in delivery area that the addition of docking stations triggers (Fig. 16), proves that this solution does do what it promises while still maintaining respectable delivery times. Perhaps, at a glance, the most surprising results may be how sensitive the delivery area can be to the drone's range in the [loaded] state (Fig. 12) since it shows how vaster the delivery area can become for modest range improvements, though its sensitivity is trivial to explain i.e. for each km extra of range improvement, several circular areas have their radius simultaneously increased which in turn increases their area. The area of a circle is quadratically dependant on its radius, and thus for a linear radius growth, there is a quadratic area growth. Increasing the range acts as a second, least demanding way of improving upon the new-found delivery area already provided by the existence of docking stations.

Analyzing the fleet dimensions behaviour (Fig. 11) shows what

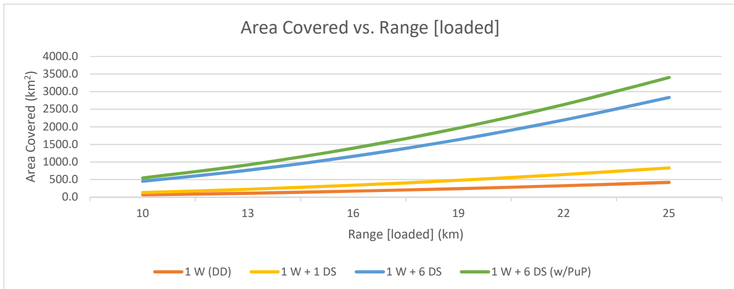


Figure 12: Area Covered vs. $R_{\text{loaded}}$.

might be one of the most overlooked and crucial aspects of implementing a network of delivery drones. Depending on how active the demand in a certain location is, the amount of drone activity in the warehouse and its proximity can become very high. This requires thoughtful infrastructure planning when dimensionalizing a network of this sort. Warehouses need to be designed and prepared so as to accommodate drone delivery to a large scale. Security issues like air traffic, collision avoidance systems, landing zones, departure pads, among others, are paramount to ensure a smooth implementation in the future.

This work also tried to expand on the way a delivery drone can operate by focusing on the fact that the drone itself changes as a vehicle mid-mission. After delivering a package, the payload decreases which equates to less weight for the electric motors to lift and, therefore, to a performance boost margin that it can use to enhance the service, hence the implementation's inclusion of both a [loaded] and [unloaded] state for the drone. Increasing the speed after delivery i.e. increasing $U_{\text{unloaded}}$ (Fig. 13) showed that it can be beneficial to make use of the performance margin gained by the weight relief provided by the delivery, albeit not as significant as one would expect.

Furthermore, depending on the weight of the package, which depends solely on the items ordered, this performance margin is going to vary accordingly, and in some cases it may actually be

| Deliveries/Hour/Drone vs. Speed [unloaded] | | | | | |
|---|---|---|---|---|---|
| Scenario | | 1 W (DD) | 1 W + 1 DS | 1 W + 6 DS | 1 W + 6 DS (w/PuP) |
| Speed [unloaded] (km/h) | 70 | 2.68 | 1.25 | 0.90 | 0.98 |
| | 80 | 2.74 | 1.28 | 0.93 | 1.00 |
| | 90 | 2.80 | 1.30 | 0.95 | 1.02 |
| | 100 | 2.84 | 1.32 | 0.96 | 1.04 |

Figure 13: Deliveries/Hour/Drone vs. $U_{\text{unloaded}}$.

detrimental to the service if, for instance, it incurs in an energy consumption rate so high that it can't ensure a safe return to the warehouse. In a real-world application, depending on the properties of the delivery drone used, it may be useful to develop an intelligent system that measures the vehicle's energy level after delivery and optimizes its performance based on that. Likewise, the vehicle's geometry may also be of important analysis since some of the current delivery drone prototypes do change their geometry by delivering the cargo box itself which can result in an aerodynamic benefit and it might be an interesting subject to study in future work.

When exploring how the charging rate can affect the effectiveness of the service, results returned show that employing fast charging methods is beneficial to the network since it improves the rate at which each drone delivers packages, though some of the values tested represent unrealistic $T_{\text{fcharge}}$ times. In a real-world application, if battery swapping is not an option, an analysis regarding charging times has to be thoroughly conducted since improving them may increase each drone's performance to a point where the network won't need as many drones to attain a certain rate of deliveries/hour/drone.

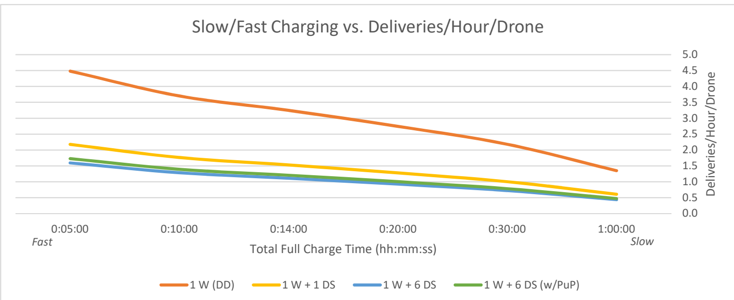


Figure 14: Slow/Fast Charging vs. Deliveries/Hour/Drone.

One of the most important proposed solutions in this work was the introduction of docking stations with the ability to act as pick-up points, mainly as a response to the large amount of potential customers that do not live in residential buildings suitable for conventional drone delivery as well as customers that are not in any location in particular but simply want to order something and get it quickly by placing the order to a nearby docking station. One of the main advantages of this solution is that it improves upon the normal concept by making use of what is already in it, i.e. the docking stations, and upgrading them to do more, to act beyond its battery-recharging functionality and potentially improve the versatility of the service in a scale that far out-weighs the extra-complexity of having to develop systems that enable them to act as pick-up points. As far as it can be understood, the PuP-enabled environment outperformed the PuP-less environment in every single result. It always returned higher deliveries/hour/drone values for the same inputs, it always presented similar or shorter operation times as well as a significantly lower average delivery time and it proved itself more accessible to more customers by being able to deliver to more demand points. In order to further demonstrate that adding a pick-up functionality can't act as a detriment to the network's performance, the test shown in Fig. 15 shows that, if anything, more deliveries using the pick-up point functionality result in higher deliveries/hour/drone. Of course, one has to be aware of the limitations of the implementation developed compared to a real-world application, but the results obtained are very positive towards the implementation of a pick-up point functionality.

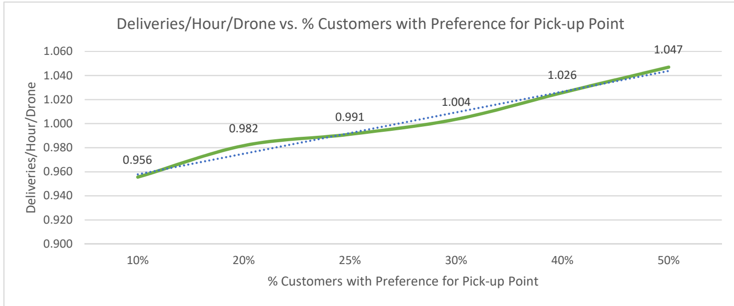


Figure 15: Deliveries/Hour/Drone vs. % Customers with Preference for Pick-up Point.

To conclude, depending on how its implementation is done, docking stations can provide incredibly satisfactory solutions for a great portion of the problems that drone delivery currently faces. The implementation developed and the results returned satisfy the motivation behind this work and show how a network of docking stations, in the context of the implementation developed, behaves. The introduction of the pick-up point functionality also proved to work exceedingly well, resulting in a better performing network overall and it can be one of the definite solutions for the specific problems it tackles.

## References


[1] Amazon. Amazon Prime Air. URL https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8node=8037720011. Accessed: 2019-02-10.

[2] S. Brar, R. Rabbat, V. Raithatha, G. Runcie, and A. Yu. Drones for Deliveries. Technical report, Berkeley Engineering, November 2015.

[3] S. Jung and H. Kim. Analysis of amazon prime air uav delivery service. *Journal of Knowledge Information Technology and Systems*, 12:253–266, 04 2017.

[4] A. M. Lawson. Consumer Behaviors of Drone Delivery, May 2017.

[5] O. Sapunkov, V. Pande, A. Khetan, C. Choomwattana, and V. Viswanathan. Quantifying the promise of beyond liion batteries. *Translational Materials Research*, 2:045002, 11 2015.

[6] A. Welch. A cost-benefit analysis of Amazon Prime Air, 2015.

[7] J. Xu. Design Perspectives on Delivery Drones. Technical report, RAND Corporation, 2017.