Proceedings of the 2020 Winter Simulation Conference
K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, eds.

CAROUSEL INSPIRED VIRTUAL CIRCULATION: A SIMULATION MODEL FOR UAV ARRIVAL AND LANDING PROCEDURE UNDER RANDOM EVENTS

Gregoire Ky
Sameer Alam
Vu Duong

Air Traffic Management Research Institute
Nanyang Technological University
65 Nanyang Drive, 637460, SINGAPORE

ABSTRACT

The current growth in the use of Unmanned Aerial Vehicles has brought to attention the need to develop a corresponding new infrastructure. One of those is the Carousel Inspired Virtual Circulation method, which consists on having Unmanned Aerial Vehicles represented as virtual blocks circulating alongside a virtual closed circuit. The purpose of this paper is to model and simulate this method in a landing configuration for large Unmanned Aerial Vehicles and evaluate its efficacy. Compared to previous works, this simulation will take into account more restrictive parameters and consider a randomized disruption, as well as an emergency landing situation. The results obtained after three runs of the simulation showed that, for each simulation, at least one virtual block landed after running out of battery. Thus, the limits of the method have been identified and further optimization of the landing sequence will be required for future works.

1 INTRODUCTION

Recent years have seen the advent and growth of unmanned aircraft systems (UAS). Drone deliveries, air taxi... the possibilities of potential applications currently envisioned are numerous and elaborate exciting prospects for the future of Urban Air Mobility (UAM). However, while the demand is gradually growing, the infrastructure enabling Unmanned Traffic Management (UTM) is still under development.

Furthermore, new concerns and perspectives regarding this development are also emphasized by respective authorities who are currently elaborating new frameworks to incentivize UTM stakeholders to seriously consider this development. For example, SESAR Joint Undertaking has launched the U-space program (SESAR Joint Undertaking 2017) to define such framework, which consists of gradually enabling services to all users while maintaining high levels of safety inside a highly digitalised environment. Following this, principles and considerations have been carefully elaborated in order to help giving guidance to research related to UAM (Thipphavong et al. 2018; SESAR Joint Undertaking 2019).

The need for such an infrastructure has been emphasized over the last decade. Indeed, Weibel and Hansman (2006) discussed considerations in this regard and proposed potential architectures while taking into account early safety concerns. More recently, NASA developed a Concept of Operations for their UTM research initiative (Kopardekar et al. 2016). Consequently, a set of principles were drafted based on earlier works on on-demand mobility (Mueller et al. 2017) which include a minimal Air Traffic Control (ATC) infrastructure, minimal workload for controllers, minimal interactions with other airspace users, compliance to safety regulations, resilience, economic efficiency regarding high-demand operations and a high flexibility of use combined with a structure change only when necessary (Thipphavong et al. 2018). As a matter of fact, newly designed structures have started to emerge. For example, the Metropolis project discussed the relationship between capacity and structure in prospective of a highly developed urban airspace

(Sunil et al. 2015). In particular, this paper introduces several concepts for future airspace structures, which are: a free flight focused mixed airspace, a layer based airspace, a zone based airspace and a tube network.

One particular area of note would be the landing procedure. In a congested airspace, several restrictions and new methods will be required in order to regulate the landing flow of vehicles. For example, Air Traffic Management has widely accepted the Point Merge method (Favennec et al. 2010) to organize the arrival flow. There is a need to explore new ideas in order to build an efficient system which complies with the principles defined in (Thipphavong et al. 2018), maintains high levels of safety, and delivers the demanded services.

The objective of this paper is to simulate the Carousel Inspired Virtual Circulation (CIVC) method for an arrival scenario for large Unmanned Aerial Vehicles (UAV) which require a runway. This method was introduced in Ky et al. (2020) and is based on the uniform movement of consecutive horses who all follow the same predefined line. As a matter of fact, the Carousel concept has already been used as a source of inspiration in diverse research areas. For example, it has been used for digital video broadcasting (Crinon 1997), as well as optimization problems by combining it with greedy algorithms (Cerrone et al. 2017). Regarding UTM, this method consists of having UAVs enter a virtual slot which will then circulate alongside a virtual closed circuit. The advantage of this approach helps combining both the landing sequencing and the hovering around several runways. In the eventuality that future droneports will each possess multiple landing points, this perspective offers an innovative way of sequencing the arrival of UAVs, which simplifies the landing procedure while keeping an harmonized traffic around the droneport. The purpose of this paper is to showcase such a possibility through a case study and consequently challenge the limits of this method by simulating it.

Indeed, this paper will present a simulation of the aforementioned CIVC to an arrival procedure with two distinct runways and run several simulations in a similar fashion to what was done in earlier works (Ky et al. 2020), which demonstrated the flexibility of the method's geometry as well as successfully simulated it for an arrival procedure on a single runway while taking into account randomized parameters, namely a randomly chosen remaining battery life and a randomized time spent on the runway. This paper will introduce new restrictions to the earlier simulation. Firstly, as opposed to the circular circuit in Ky et al. (2020), the geometric configuration will take into account two runways and consequently, the circuit will be designed by a piecewise arc. Figure 1 illustrates the geometrical design that was chosen for this purpose. Furthermore, additional random-based restrictions will also be added. Indeed, a disruption will be simulated by randomly reducing the overall circulation speed for short time intervals. Another restriction which will be considered is the emergency landing of a randomly chosen vehicle, at a randomly chosen time. Those new restrictions were chosen in order to simulate a more realistic scenario which will challenge the limits of the CIVC method. Section 2 will first provide an extensive analytical description of the geometry, and then describe the randomized events chosen. Thus, Section 3 will go through the details of the simulations as well as illustrate the results obtained. Finally, Section 4 will discuss the possibilities opened by the method and the limits introduced by it, before giving a few ideas regarding its compatibility with methods that either already exist, or are in the development pipeline.

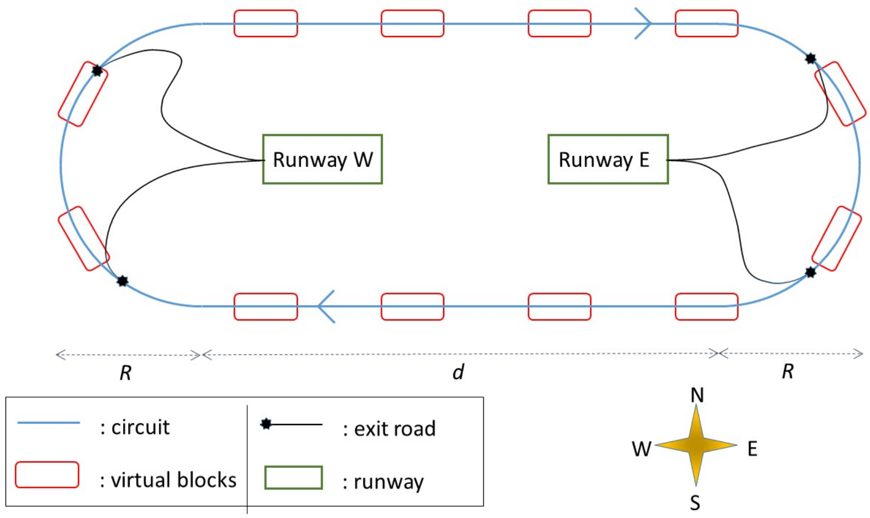

Figure 1: Conceptual depiction of the design chosen for the case study.

2 DESCRIPTION OF THE PROBLEM

As the demand grows, the airspace capacity needs to adapt. As such, a flexible airspace structure is required. Consequently, several studies have applied shortest path planning algorithms to a UAV oriented airspace and simulated different methods for conflict detection and resolution (Chakrabarty and Ippolito 2020; Ho et al. 2018; Tan et al. 2019). On the other hand, Bosson and Lauderdale (2018) simulated a structure more similar to current ATM in which the routes are directly linking each landing points to each other. However, in the eventuality of a dense traffic around a single droneport, especially in the presence of large UAVs which require a runway, a structure will be needed in order to provide a landing sequence and regulate the arrival flow. The CIVC is a potential idea in order to do so.

Air Traffic Management is a safety-critical environment and that extends to UTM as well. As such, before any field trial, carefully planed simulations are required to prove feasibility. Furthermore, the CIVC was originally designed to be operated in a highly automated environment and thus, the concept is not mature enough to be eligible for a field trial. Consequently, this paper focuses on the simulation of the CIVC in order to study its current efficacy under specific constraints and identify its potential early limits.

Complementary to what was done in Ky et al. (2020), this study will take into account a more complex environment. Indeed, the first consideration will be to have two runways instead of one, thus enabling the simultaneous landing of two virtual blocks. Furthermore, the case study in Ky et al. (2020) considers a circuit made of a single circle, while this study considers a more complex design, which is built using a piecewise parameterized arc. Subsection 2.1 will thus describe this new circuit in both a geometrical and analytical way, while Subsection 2.2 will switch from a geometrical parameter to a time perspective in order to implement two random events, which are a general disruption and an emergency landing.

2.1 Geometric Design

The new circuit considered for this study will be divided into four different parts, two half circles with the same radius, and two parallel lines of the same length. Figure 2 illustrates this circuit.

In order to simplify the description of this circuit for later on, compass directions will be used to identify the different parts and coordinates. Consequently, part 1 from (x_W, y_S) to (x_W, y_N) will be the western circle, part 2 from (x_W, y_N) to (x_E, y_N) the northern line, part 3 from (x_E, y_N) to (x_E, y_S) the eastern circle and part 4 from (x_E, y_S) to (x_W, y_S) the southern line. All coordinates mentioned in this paper will be described in distance terms. It is also worthwhile to note that both runways and the center of both circles were chosen

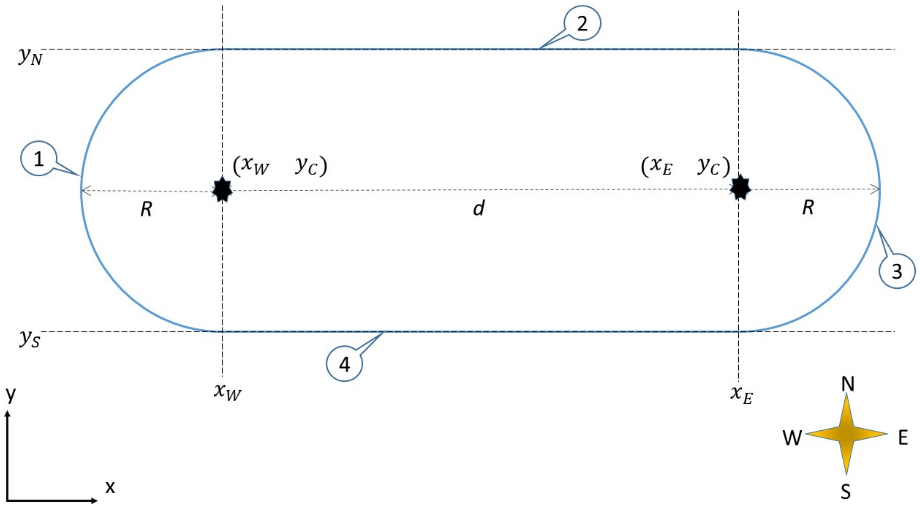

Figure 2: Geometrical description of the circuit.

to be at the same y-coordinate y_C. Thus the following relations between the coordinates: $d = x_E - x_W$, $2R = y_N - y_S$, $y_C = y_S + R = y_N - R$

Additionally, the starting point of the arc $(u = 0)$ is chosen to be at the southwestern point (x_W, y_S) and the blocks are directed to follow the circuit in a clockwise fashion. Considering all of those coordinates, it is now possible to write the analytical description of each arc. However, in order to obtain a continuity regarding the geometric parameter, a variable is necessary in order to translate the original formulations to the wanted interval. Furthermore, in order to concretise the periodicity of the circuit, the geometric parameter u has been fixed to the interval [0,2π]. Thus the analytical expression of the parameterized arc:

$$\begin{pmatrix} x(u) = x_W + R\cos\left(-2u - \frac{\pi}{2}\right) \\ y(u) = y_C + R\sin\left(-2u - \frac{\pi}{2}\right) \end{pmatrix} for \quad 0 \leq u \leq \frac{\pi}{2} \quad \text{(western circle)} \tag{1}$$

$$\begin{pmatrix} x(u) = \frac{2}{\pi}(\pi - u)x_W + \frac{2}{\pi}(u - \frac{\pi}{2})x_E \\ y(u) = y_N \end{pmatrix} for \quad \frac{\pi}{2} \leq u \leq \pi \quad \text{(northern line)} \tag{2}$$

$$\begin{pmatrix} x(u) = x_E + R\cos\left(-2u + \frac{\pi}{2}\right) \\ y(u) = y_C + R\sin\left(-2u + \frac{\pi}{2}\right) \end{pmatrix} for \quad \pi \leq u \leq \frac{3\pi}{2} \quad \text{(eastern circle)} \tag{3}$$

$$\begin{pmatrix} x(u) = \frac{2}{\pi}(2\pi - u)x_E + \frac{2}{\pi}(u - \frac{3\pi}{2})x_W \\ y(u) = y_S \end{pmatrix} for \quad \frac{3\pi}{2} \leq u \leq 2\pi \quad \text{(southern line)} \tag{4}$$

Finally, the four exit roads connecting the circuit were computed using a third-degree polynomial interpolation between the access points to the runways, and four points on the circuit which were chosen at the parametric positions: $u = \frac{\pi}{8}$ (southwest exit),$\frac{3\pi}{8}$ (northwest exit),$\frac{9\pi}{8}$ (northeast exit) and $\frac{11\pi}{8}$ (southeast exit). Furthermore, the simulation will also take into account an altitude z_C at which the circuit will stay constant.

2.2 Randomized Events

Now that the circuit has been built, the randomized events can be defined. The first step subsequently required is to switch from the geometric parameter to a time perspective. Alongside the two new events, the remaining battery and the time spent on the runway from Ky et al. (2020) will also be retained. The remaining battery will be described as the remaining time before running out. Since the circuit is composed of two half circles with the same radius, as well as two straight lines of the same length, the total arc length of the circuit can be directly deduced: $L_C = 2\pi R + 2d$.

In a similar way to what has been done in Ky et al. (2020), it is necessary to introduce the speed v that each virtual block is circulating at. The total time required to travel across the whole circuit can then be obtained: $t_{total} = \frac{L_C}{v}$.

The next step consists of fractionating the arc length formula (Weisstein 2010). However, since the circuit is composed of four different pieces, it is necessary to separate the cases depending on the position. From a time perspective, it is thus required to introduce partial times representing the time to travel across one of the circuit's piece at speed v. Since the circuit is composed of two similar straight lines and two similar half circles, the subsequent times can be deduced: $t_{circle} = \frac{\pi R}{v}$, $t_{line} = \frac{d}{v}$, $t_{total} = 2t_{circle} + 2t_{line}$

The final relationship between the geometric parameter u and the time t, can then be obtained using the fractionated arc length formula from Ky et al. (2020):

$$u = \frac{L_C}{2R}\frac{t}{t_{total}} \quad \text{for} \quad 0 \leq t < t_{circle} \tag{5}$$

$$u = \frac{\pi}{2d}\left(L_C\frac{t}{t_{total}} - \pi R + d\right) \quad \text{for} \quad t_{circle} \leq t < t_{circle} + t_{line} \tag{6}$$

$$u = \frac{1}{2R}\left(L_C\frac{t}{t_{total}} + \pi R - d\right) \quad \text{for} \quad t_{circle} + t_{line} \leq t < 2t_{circle} + t_{line} \tag{7}$$

$$u = \frac{\pi}{2d}\left(L_C\frac{t}{t_{total}} - 2\pi R + 2d\right) \quad \text{for} \quad 2t_{circle} + t_{line} \leq t < t_{total} \tag{8}$$

It is worthwhile to note that those equations are restricted to the time interval $[0, t_{total}]$. As such, there is a need to adjust the time by changing it from t to t modulo t_{total}.

Now that the problem formulation has been switched to a time perspective, the implementation of randomized events in the simulation can be discussed. The first event will represent a disruption, which has been modeled by reducing the overall circulation speed. Indeed, a random percentage will be chosen every five minutes, which will then modify the speed subsequently. Consequently, all time variables in equations (5) to (8) will be numerically altered.

Algorithm 1: Pseudo-code process for the emergency landing

Index number of virtual block requiring an emergency landing: i_e;
Exit closest to virtual block i_e: e_j;
Index number of virtual block currently in a landing procedure: i_l;
if *Block i_l on the circuit* **then**
  Cancel clearance for block i_l;
  Block i_l to continue circulation;
  Block i_e cleared for landing at exit e_j;
else if *Block i_l on an exit road* **then**
  Block i_e to continue circulation until block i_l reaches the runway;
  if *Block i_e bypasses exit e_j* **then**
    $e_j = e_{j+1}$
  end
else if *Block i_l on the runway* **then**
  Block i_e cleared for landing at exit e_j;
end
Result: Virtual block i_e requiring an emergency landing has successfully landed

The second event which will be taken into account for the simulations is the addition of an emergency situation, in which one randomly chosen virtual block will, at a randomly chosen time stamp, enter into a critical state, thus requiring immediate landing clearance. Algorithm 1 describes the programmed procedure in this case.

The time stamp at which Algorithm 1 triggers is randomly chosen first. Once reached, the block to have an emergency landing (i_e) will also be randomly chosen among the remaining blocks at that time. Furthermore, Algorithm 1 has been written while only taking a single runway into account. In the eventuality that the emergency landing is delayed enough for the runway to change, Algorithm 1 will be applied in the same way, with the exception of i_l which will be modified consequently.

3 THE SIMULATION

Considering everything that has been mentioned in Section 2, all the required elements are now prepared for the simulations. The landing sequence itself has been defined in Ky et al. (2020) and consists of first identifying the vehicle with the least remaining battery time, then proceed to the closest exit point and follow the exit road until the corresponding runway, on which the vehicle will spend a previously randomly allocated time. Figure 3 depicts a flowchart explaining how the landing sequence has been programmed for a single runway simulation.

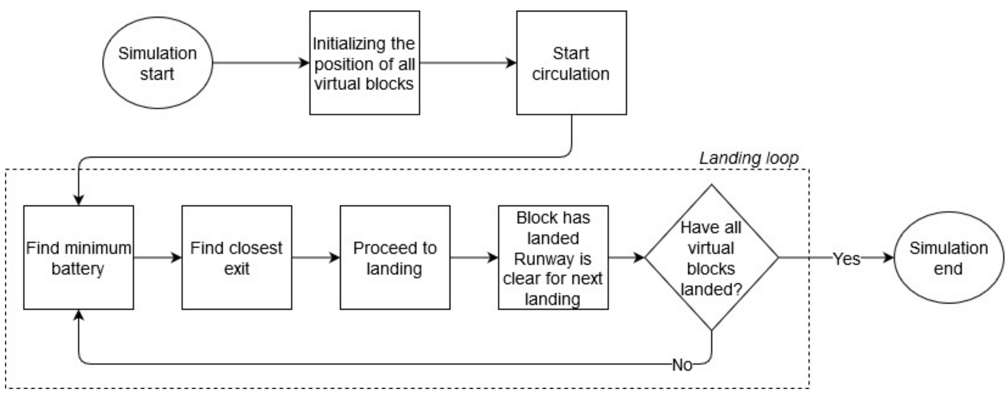

Figure 3: Landing process for a single runway.

A virtual block is defined as a three dimensional rectangle whose center is located on the circuit at all time and has one centerline tangent to the circuit at the center. Each of their respective positions are initially computed using the geometric parameter $u = \frac{2\pi}{N_b}k$ for virtual block k, with $1 \leq k \leq N_b - 1$. The extension of Figure 3 to a two runways design is obtained by separating the virtual blocks in two categories, depending on which runway is the closest to each block according to their positions on the circuit. For the eastern runway, it concerns the virtual blocks whose geometric parameter is within the interval $u \in]\frac{3\pi}{8}, \frac{11\pi}{8}]$, while for the western runway, the virtual blocks have their geometric parameter in $u \in [0, \frac{3\pi}{8}] \cup]\frac{11\pi}{8}, 2\pi]$. After this separation, the process is exactly the same as in Figure 3.

Putting Figure 3 into perspective, the landing process can thus be decomposed into two intertwined functions: the clearance providing, as well as the virtual block landing trajectory. The former being a simplified single-server queuing system, as presented in Law and Kelton (2000), while the latter focuses on the trajectory of each virtual blocks.

This trajectory function operates depending on where the virtual blocks are located. If on the circuit, the function uses equations (1) to (4), a third degree polynomial interpolation for the four exit roads, and a linear time-decrease for the runway depending on the randomly allocated time spent on the runway. Furthermore, the function also takes into account two transitions, one from the circuit to the exit roads, and one from the exit roads to the runway. Both transitions occur when the virtual block cleared for landing reaches the vicinity of the transit point. The first transition (circuit to exit road) will trigger once the virtual block enters a vicinity sphere centered around the exit point with a radius arbitrarily chosen at 5 m. The second transition (exit road to runway) happens in the same fashion as the first one, the difference being the radius of the vicinity sphere which has been arbitrarily fixed at 2m.

Additionally to the random elements introduced in Subsection 2.2, the ones from Ky et al. (2020), such as the time spent on the runway and the remaining battery, will also be kept. Remaining battery for each virtual block will be randomly chosen between 15 min and 90 min, while the time spent on the runway will be randomly chosen between 30 s and 5 min for each virtual block. The battery consumption rate is supposed fixed. Those random variables were further restrained in order to challenge the limits of the CIVC method. All random elements are generated using MATLAB's *rand* function, which is a uniform random distribution whose default random number generator is a Mersenne twister.

The geometric parameters were then selected as such:

- $x_W = 40$ m
- $x_E = 120$ m
- $y_S = 10$ m
- $y_N = 70$ m
- Access point to the western runway: $(30 \text{ m}, 40 \text{ m})$
- Access point to the eastern runway: $(130 \text{ m}, 40 \text{ m})$

Considering those values, the other geometric parameters can thus be deduced:

- $R = 30$ m
- $d = 80$ m
- $L_C = 348.5$ m
- $y_C = 40$ m

The vicinity of the transition points has been defined as the sphere centered at the transition point in which the virtual block switches between two distinct parts of the circuit. The radius of such sphere has been fixed at 5 m for the circuit to exit road transition, and 2 m for the exit road to runway transition.

Furthermore, the number of virtual blocks was arbitrarily fixed at $N_b = 15$, with a block length of 5 m and a separation length of 19.5 m (Ky et al. 2020). Consequently, it is possible to write an analytical formula (9) which describes the position of each virtual blocks, at all time, and is directly computed from equations (5) to (8):

$$u_{i_b} = u\left(t + (i_b - 1)\frac{2\pi}{N_b}\right) \quad \text{for} \quad i_b = 1 \ldots N_b \tag{9}$$

Additionally, the simulation will also take into account an altitude $z_C = 50$ m above ground level for the circuit. All of those elements were implemented on MATLAB in order to run a simulation. Figure 4 represents the three dimensional considered circuit and the chosen repartition of virtual blocks as it results from the program.

Up to this point, the simulation will flow in a similar manner than the one in Ky et al. (2020). Compared to the results obtained in Ky et al. (2020), the implementation of two runways instead of one enables the simultaneous landing of two virtual blocks. The next step will be to implement the additional random events. However, the implementation of a randomly generated disruption brings an additional constraint which will greatly slow down the whole process. Figure 5 shows the generated disruptions translated into a speed decrease percentage.

Furthermore, as can be inferred from equations (5) to (8), this speed decrease will directly influence the time required to travel across a single lap t_{total}, which will then directly change the relation between the

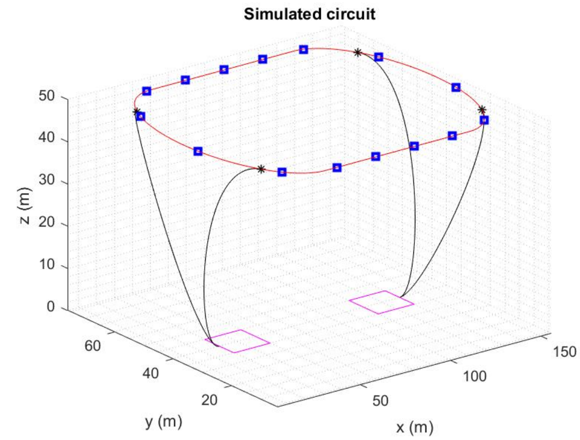

Figure 4: Three dimensional representation of the simulated circuit.

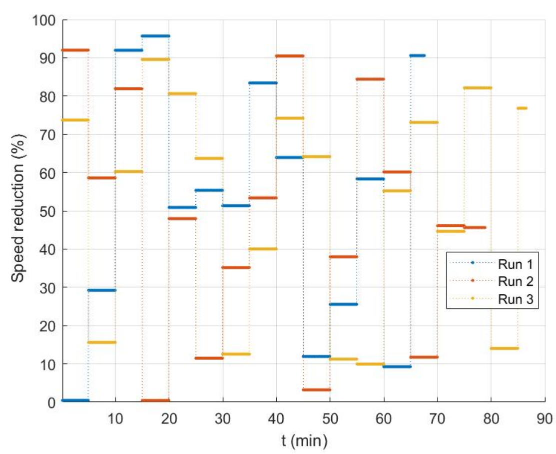

Figure 5: Distribution of the disruption computed for each run.

geometric parameter u and the time t. Figure 6 displays this subsequent change for the three simulations, by computing the new profiles for each change.

In order to maintain the continuity of those profiles, an offset was added at each change. Those two figures put into perspective the impact the modeled disruptions cause on the whole process. Indeed, the average speed decrease and standard deviation for each run is:

1. 50.12 %, $\sigma = 0.3066$, for a total running time of 1 h 07 min 34 s
2. 52.43 %, $\sigma = 0.2946$, for a total running time of 1 h 18 min 52 s
3. 48.68 %, $\sigma = 0.2745$, for a total running time of 1 h 26 min 31 s

The trivial conclusion from those values combined with both Figures 5 and 6 is that the greater the speed decrease is, the longer it will take for all blocks to land. It is also worthwhile to note that those three simulations also took into account the emergency landing event.

To identify the landing sequence, the program simply detects the virtual block with the least remaining battery and provides a clearance depending on which runway is the closest to its current position, while taking into account the direction of the circulation: $u \leq \frac{3\pi}{8}$ and $u > \frac{11\pi}{8}$ for the western runway, $u \leq \frac{11\pi}{8}$

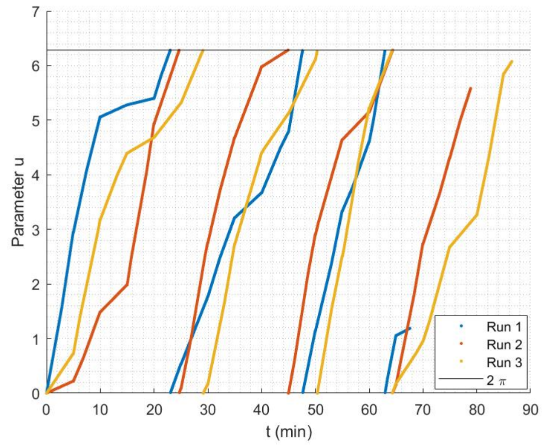

Figure 6: Distribution of the disruption computed for each run.

and $u > \frac{3\pi}{8}$ for the eastern runway. The clearance for landing is only provided when there are no other virtual blocks on both the corresponding runway and exit roads. Considering all those settings, Tables 1-3 compiles all the initial and resulting time data for each respective simulation. In each table, the virtual block which was randomly selected for the emergency procedure is written in read. For each simulation, the virtual block which had an emergency landing is:

1. Virtual block 11 after 38 min 06 s
2. Virtual block 14 after 25 min 42 s
3. Virtual block 15 after 27 min 57 s

Table 1: Initial and resulting times for the first simulation.

Result time (all times are in hh:mm:ss format.)				
Index number of virtual block	*Remaining battery expectancy at starting time*	*Remaining battery expectancy after landing*	*Time spent on the runway*	*Elapsed time between starting time and landing*
1	00:36:07	00:06:16	00:48	00:28:33
2	00:15:31	00:08:27	02:20	00:06:45
3	00:52:11	00:12:40	01:34	00:37:48
4	01:29:08	00:18:46	02:42	01:07:18
5	01:10:20	00:07:15	04:07	01:00:22
6	00:38:18	00:23:34	02:12	00:14:06
7	01:00:01	00:10:03	02:50	00:47:48
8	01:13:37	00:09:54	00:56	01:00:58
9	00:23:21	00:15:22	04:35	00:07:39
10	00:58:26	00:30:29	01:26	00:26:45
11	00:26:01	01:44:12	02:13	00:51:55
12	01:06:44	00:00:00	03:28	01:07:34
13	00:33:13	00:16:30	03:54	00:16:00
14	00:40:42	00:03:54	01:16	00:35:12
15	00:55:54	00:00:00	02:49	00:55:13

Table 2: Initial and resulting times for the second simulation.

Result time (all times are in hh:mm:ss format.)				
Index number of virtual block	***Remaining battery expectancy at starting time***	***Remaining battery expectancy after landing***	***Time spent on the runway***	***Elapsed time between starting time and landing***
1	01:19:34	00:00:00	03:21	01:18:52
2	00:43:28	00:10:39	04:09	00:31:24
3	01:08:24	00:00:00	02:31	01:08:10
4	00:54:16	00:02:40	04:14	00:49:21
5	00:42:16	00:23:55	01:04	00:17:33
6	00:47:36	00:05:19	02:49	00:40:27
7	01:06:34	00:10:44	03:43	00:53:25
8	00:32:01	00:20:09	01:37	00:11:21
9	01:28:26	00:14:21	02:53	01:10:52
10	01:28:11	00:30:15	02:13	00:55:25
11	00:36:43	01:44:12	04:06	00:29:12
12	00:40:23	00:06:11	03:31	00:21:39
13	01:29:44	00:09:23	04:55	01:16:52
14	01:14:10	00:30:50	04:43	00:41:27
15	01:14:37	00:10:35	03:06	01:01:16

Table 3: Initial and resulting times for the third simulation.

Result time (all times are in hh:mm:ss format.)				
Index number of virtual block	***Remaining battery expectancy at starting time***	***Remaining battery expectancy after landing***	***Time spent on the runway***	***Elapsed time between starting time and landing***
1	00:23:11	00:00:00	04:47	00:58:31
2	01:02:36	00:00:00	02:39	01:09:46
3	00:21:04	00:12:42	03:13	00:08:00
4	00:45:51	00:03:39	03:10	00:40:21
5	01:08:27	00:00:00	01:31	01:17:31
6	00:22:32	00:09:12	03:30	00:12:45
7	01:15:45	00:00:00	01:12	01:26:31
8	01:02:50	00:01:59	03:59	00:58:13
9	01:22:21	00:09:32	01:28	01:09:40
10	01:01:38	00:17:25	01:16	00:42:18
11	00:46:06	00:30:12	03:46	00:15:12
12	01:03:34	00:11:08	01:39	00:50:10
13	00:51:42	00:19:55	01:14	00:30:24
14	00:11:02	00:00:07	03:19	00:20:57
15	01:02:48	00:24:51	01:38	00:36:18

An interesting perspective from the result tables is that for each simulation, at least two of the vehicles ran out of battery. It can thus be inferred that further refinement will be needed in future works.

4 DISCUSSION AND FURTHER CONSIDERATIONS

The objective of this paper was to apply the CIVC method to a more complex configuration than what was done in Ky et al. (2020), while adding further constraints in the form of randomized events. Furthermore, restraining the battery range definitely assisted in identifying the limits of the method. Although the initial results obtained at this point were not satisfactory from an operational perspective, several points can already be considered for future developments. The fact that several virtual blocks systematically ran out of battery in each simulation proves that there is an optimization problem in the process.

Two potential directions stand out from this point. The first one relies on the relation between capacity and efficiency of the method and more particularly, the impact the geometrical configuration has on the problem. Considering the results obtained in Section 3, it can be discussed that, for the same amount of virtual blocks, a smaller circuit or an additional runway could guarantee the safe landing of all virtual blocks. An interesting approach for future work would thus be one based on the relation between the circuit's capacity and the geometrical configuration and its optimization.

On the other hand, the second guideline lies on the sequencing program itself. Indeed, the approach used in this study only makes use of MATLAB's default minimum function. A proper sequencing algorithm which could implement, for example, data driven techniques, thus greatly enhancing the efficiency of the method.

An important consideration of this study is regarding the finite aspect of the simulation, which proceeds for the landing of a fixed number of vehicles. Future work should take into account the addition of new vehicles onto the circuit while the system is still operating, thus enabling a simulation for a continuous flow of traffic. For example, a comparison can be made with the Point Merge method (Favennec et al. 2010). While both methods share a few similarities on generating a landing sequence, they are also fairly different. On one hand, the CIVC has a high geometric flexibility and thus easily enables landing on multiple runways. On the other hand, while it is also possible to have multiple runways with the Point Merge method (Liang et al. 2017), its main advantage over the CIVC comes from enabling the arrival of a continuous traffic flow. However, those two methods may actually be compatible with each other. Indeed, a potential idea combining the two methods for UTM would be to have Point Merge sequence the UAVs onto the CIVC circuit, which will then dispatch the vehicles onto the multiple runways of, for example, a drone port the circuit is surrounding. In this configuration, the Point Merge would then become the link connecting the drone port to the airspace, while the CIVC would thus be the UTM equivalent of the Airport Surface Movement.

To conclude, this paper illustrates the capabilities of the CIVC method, while also finding its limits. Consequently, while there are still many challenges to overcome, tentative prospects for the development of the method have been identified. Even if those initial results did not give complete satisfaction, they nevertheless provided real progress and new perspectives towards the future of the CIVC, as well as possibly towards the future of Urban Air Mobility.

ACKNOWLEDGMENTS

This research is supported by the Civil Aviation Authority of Singapore and Nanyang Technological University, Singapore under their collaboration in the Air Traffic Management Research Institute. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of the Civil Aviation Authority of Singapore.

REFERENCES

Bosson, C., and T. Lauderdale. 2018. "Simulation Evaluations of an Autonomous Urban Air Mobility Network Management and Separation Service". In *Aviation Technology, Integration, and Operations Conference*. June 25th-29th, Atlanta, Georgia, 3365.

Cerrone, C., R. Cerulli, and B. Golden. 2017. "Carousel Greedy: a Generalized Greedy Algorithm with Applications in Optimization". *Computers & Operations Research* 85:97–112.

Chakrabarty, A., and C. Ippolito. 2020. "Autonomous Flight for Multi-copters Flying in UTM-TCL4+ Sharing Common Airspace". In *AIAA Scitech Forum*. January 6th-10th, Orlando, Florida, 0881.

Crinon, R. J. 1997. "The DSM-CC Object Carousel for Broadcast Data Services". In *IEEE International Conference on Consumer Electronics*. June 11th-13th, Rosemont, Illinois, 246-247.

Favennec, B., F. Vergne, and K. Zeghal. 2010. "Point Merge Integration of Arrival Flows Enabling Extensive RNAV Application and Continuous Descent–Operational Services and Environment Definition". Technical report, Aufl. Brétigny, Eurocontrol Experimental Centre, Brétigny-sur-Orge, France.

Ho, F., R. Geraldes, A. Goncalves, M. Cavazza, and H. Prendinger. 2018. "Improved Conflict Detection and Resolution for Service UAVs in Shared Airspace". *IEEE Transactions on Vehicular Technology* 68(2):1231–1242.

Kopardekar, P., J. Rios, T. Prevot, M. Johnson, J. Jung, and J. Robinson. 2016. "Unmanned Aircraft System Traffic Management (UTM) Concept of Operations". In *16th AIAA Aviation Technology, Integration, and Operations Conference*. June 13th-17th, Washington, D.C., 3292.

Ky, G., S. Alam, and V. Duong. 2020. "A Traffic Method for Unmanned Aircraft Systems on a Virtual Closed Circuit". In *Conference on Artificial Intelligence and Data Analytics in Air Transportation*. February 3rd-4th, Singapore.

Law, A. M., and W. D. Kelton. 2000. *Simulation Modeling & Analysis*. 3rd ed. New York: McGraw-Hill, Inc.

Liang, M., D. Delahaye, and P. Maréchal. 2017. "Integrated Sequencing and Merging Aircraft to Parallel Runways with Automated Conflict Resolution and Advanced Avionics Capabilities". *Transportation Research Part C: Emerging Technologies* 85:268–291.

Mueller, E., P. Kopardekar, and K. Goodrich. 2017. "Enabling Airspace Integration for High-Density On-Demand Mobility Operations". In *17th AIAA Aviation Technology, Integration, and Operations Conference*. June 5th-9th, Denver, Colorado, 3086.

SESAR Joint Undertaking 2017. "U-space - Blueprint". https://www.sesarju.eu/sites/default/files/documents/reports/U-space%20Blueprint%20brochure%20final.PDF, accessed 5th March 2020.

SESAR Joint Undertaking 2019. "Initial View on Principles for the U-space Architecture". https://www.sesarju.eu/sites/default/files/documents/u-space/SESAR%20principles%20for%20U-space%20architecture.pdf, accessed 10th March 2020.

Sunil, E., J. Hoekstra, J. Ellerbroek, F. Bussink, D. Nieuwenhuisen, A. Vidosavljevic, and S. Kern.. 2015. "Metropolis: Relating Airspace Structure and Capacity for Extreme Traffic Densities.". In *11th USA/EUROPE Air Traffic Management R&D Seminar*. June 23rd-26th, Lisbon, Portugal, 1-10.

Tan, Q., Z. Wang, Y. Ong, and K. Low. 2019. "Evolutionary Optimization-based Mission Planning for UAS Traffic Management (UTM)". In *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. June 11th-14th, Atlanta, Georgia, 952-958.

Thipphavong, D., R. Apaza, B. Barmore, V. Battiste, B. Burian, Q. Dao, M. Feary, S. Go, K. Goodrich, J. Homola, H. Idris, P. Kopardekar, J. Lachter, N. Neogi, H. Ng, R. Oseguera-Lohr, M. Patterson, and S. Verma. 2018. "Urban Air Mobility Airspace Integration Concepts and Considerations". In *Aviation Technology, Integration, and Operations Conference*. June 25th-29th, Atalanta, Georgia, 3676.

Weibel, R., and R. Hansman. 2006. "Safety Considerations for Operation of Umanned Aerial Vehicles in the National Airspace System". Technical Report ICAT-2005-1, MIT International Center for Air Transportation, Cambridge, Massachusetts.

Weisstein, E.W. 2010. "Arc Length. From MathWorld-A Wolfram Web Resource". http://mathworld.wolfram.com/ArcLength.html, accessed 16th March 2020.

AUTHOR BIOGRAPHIES

GREGOIRE ARTHUR KY is a Research Associate at the Air Traffic Management Research Institute of Nanyang Technological University, Singapore. He obtained a Master of Engineering degree from the Ecole Nationale Superieure d'Electricite et Mecanique (ENSEM) in 2017 as well as a Master of Science from the Hong Kong University of Science and Technology in 2018. His research interest include mechanical simulations as well as quantum computing, and their potential applications to air traffic management. His email address is ga.ky@ntu.edu.sg.

SAMEER ALAM is the Programme Director of AI & Data Analytics at the Air Traffic Management Research Institute of Nanyang Technological University, Singapore. He obtained PhD in Computer Science in 2008 specializing in Artificial Intelligence from University of New South Wales (UNSW), Australia and M.Tech. in Computer Science in 1999 from BIT, Mesra, India. His post-doctoral research, on developing Artificial Intelligence algorithms for future air traffic concept evaluation, was sponsored by EUROCONTROL (2009-2010) and Air Services Australia (2010-2011). His research interest are in Artificial Intelligence and Machine Learning algorithms applied to Air Traffic Management and Airport Operations. His e-mail address is sameeralam@ntu.edu.sg.

VU DUONG is Director of Air Traffic Management Research Institute of Nanyang Technological University, Singapore. He studied at Ecole Nationale des Ponts et Chaussées of France where he obtained the degree of Master in Engineering in 1986, and Ph.D. in Artificial Intelligence in 1990. He is a recognized leader in Air Traffic Management research. During his service time with EUROCONTROL (1995-2012) he had contributed to the advances of air transportation that included research that have led to Autonomous Aircraft Operations, Sector-less Air Traffic Management, Trajectory-based Control, and numerous work on the foundation of Air Traffic Flow Management. His email address is vu.duong@ntu.edu.sg.