A traffic method for unmanned aircraft systems on a virtual closed circuit

Ky, Gregoire Arthur; Alam, Sameer; Duong, Vu Nguyen

2020

Ky, G. A., Alam, S. & Duong, V. N. (2020). A traffic method for unmanned aircraft systems on a virtual closed circuit. 2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT).

<https://dx.doi.org/10.1109/AIDA-AT48540.2020.9049166>

<https://hdl.handle.net/10356/147692>

<https://doi.org/10.1109/AIDA-AT48540.2020.9049166>

© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at:
<https://doi.org/10.1109/AIDA-AT48540.2020.9049166>

Downloaded on 26 Apr 2023 22:55:06 SGT

A traffic method for unmanned aircraft systems on a virtual closed circuit

Introducing the Carousel method

Gregoire Arthur Ky, Sameer Alam, Vu Nguyen Duong
Air Traffic Management Research Institute
Nanyang Technological University
Singapore
ga.ky@ntu.edu.sg

Abstract—This paper introduces a new traffic method for unmanned aircraft systems traffic management, in terminal maneuver space, called the Carousel method. It revolves around the circulation of virtual blocks alongside a closed circuit. This paper emphasizes on the versatility of this method and showcases a simulation of one possible application to an operational scenario, as well as discussing further enhancements for the future of the method.

This paper demonstrates the effectiveness of the geometric flexibility of the Carousel method. It first proves the entanglement between all its geometrical considerations, namely the separation length, length of the virtual blocks and maximal number of blocks on the circuit, as well as proving its geometrical flexibility through a series of simulations. Then, it successfully applies the method to a typical arrival scenario for unmanned aircraft systems, while taking into account randomized parameters, such as remaining battery and landing time, applied to every vehicle on the circuit.

Index Terms—drone traffic management (DTM), parameterized geometry, automated traffic, virtualization,

I. INTRODUCTION

In recent years, the unmanned aircraft systems (UAS) market has been constantly growing and is expected to continue. In order to cope with this evolution, SESAR Joint Undertaking (SJU) has launched the U-space program, whose objective is to formulate and design a new efficient infrastructure for all drone operations [1], while maintaining high levels of safety and avoiding hazardous conflicts with other airspace users. This vision relies on the implementation of features with high levels of automation and consequently incentivize stakeholders to look for new innovative ways of enabling it. In parallel to the current framework taking its inspiration from the Air Traffic Management (ATM) architecture, this paper suggests and introduces a different approach, which takes its roots on the automated synchronization of airspace users, as well as the virtualization of the aerial circuit they are following. This paper proposes a method for arrival sequencing for UAS at an airport. This paper envisions a scenario with UAS operating (landing and take-off) at an airport, on UAS-only-runways, similar to General Aviation only runways at many US airports. Rather than redesigning an already existing infrastructure, the

purpose of this paper is to present a new method, which can be applied in an existing Terminal Maneuver Airspace (TMA). This perspective proposes an innovative way of organizing traffic, which can later on create alternatives for U-space advanced services and beyond, in particular for features related to tactical deconfliction and dynamic capacity management [2] at airports supporting UAS operations.

Indeed, the method introduced in this paper consists of considering a close circuit, which is a continuous geometrical circuit whose starting and ending points are at the same location, thus creating a loop. To this circuit is associated a finite number of access and exit points, a configuration similar to a ring road. Then, a fixed number of virtual blocks are dispatched on the circuit, all separated by the same length and all moving alongside the circuit at the same speed. Taking aside the virtuality of this concept, the main idea comes from the synchronized circulation of the blocks alongside the circuit, like the horses on a carousel, thus the name of the method. Fig. 1 depicts the method.

Fig. 1. Conceptual depiction of the carousel method.

Since it describes an organized traffic method for several users at the same time, it is thus different from dynamic single path optimization processes such as in [3], [4] and consequently displays a different perspective for UAS operations. On the other hand, it is also possible to reference this method to the taxonomy presented in [5]. Indeed, this

approach relies on a collaborative data sharing service between all users on the circuit (distributed-dependent surveillance). Furthermore, since all vehicles are synchronized with the virtual blocks, it is consequently implied that all users must follow the same flight methodology in order to maintain separation standards (implicitly coordinated avoidance). Since separation is provided at all time, maneuvering will only apply to potential, unwanted obstacles, thus requiring maneuvers at a tactical level. Finally, since the Carousel method requires a high level of connectivity between users, it was thus originally designed for autonomous systems. The research regarding this aspect for the application of the method is out of the scope of this study.

At first, this paper will define the mathematical background behind this method, followed by a few numerical simulations to illustrate its geometric flexibility. Next, a case study will be proposed in order to apply the method to a typical TMA scenario, which will also take in consideration the remaining batteries of the vehicles. Finally, a discussion over the results of the case studies as well as an overview of potential outcomes and further considerations will be presented.

II. THE CAROUSEL METHOD

In order to visualize the capabilities of the Carousel method, there is first a need to lay down its mathematical foundation. Consequently, this section will be divided in two parts. The first part will focus on the analytical description of the method, while the second one will study its geometrical flexibility, by running a series of simulations.

A. Analytical Description

First, it is necessary to describe the closed circuit itself. In order to allow the circuit to be flexible, a three dimensional periodic parametric arc was chosen:

$$f : u \rightarrow (x(u), y(u), z(u)) \quad (1)$$

$$\exists u_p \mid \forall u \geq u_p, f(u + u_p) = f(u) \quad (2)$$

Since f is periodic, only the interval $[0, u_p]$ will be considered. Furthermore, the total arc length of the circuit is defined as follows [6]:

$$L_C = \int_0^{u_p} \sqrt{\left(\frac{dx}{du}\right)^2 + \left(\frac{dy}{du}\right)^2 + \left(\frac{dz}{du}\right)^2} du \quad (3)$$

Then, let p be the number of access points, and q the number of exit points, with their subsequent coordinates. The corresponding times of passage are defined as follows:

$$\forall i, 1 \leq i \leq p, \exists u_{a,i} \mid f(u_{a,i}) = a_i \quad (4)$$

$$\forall j, 1 \leq j \leq q, \exists u_{e,j} \mid f(u_{e,j}) = e_j \quad (5)$$

With a_i and e_j respectively, the coordinates of the access point indexed i and the coordinates of the exit point indexed

j . Now that the circuit has been defined, a certain number of assumptions need to be initially considered regarding the virtual blocks:

- a) There is a fixed number of virtual blocks n_b
- b) All blocks have the same length L_b
- c) All blocks are arranged alongside the circuit, separated by the same separation length L_s

The new lengths L_b and L_s will be considered as curvilinear lengths, alongside the parametric arc, in order to simplify further computation. Consequently, the following relation is obtained:

$$L_C = n_b * L_b + (n_b - 1) * L_s \quad (6)$$

Regarding (6): L_C only depends on the parametric arc and L_s is fixed by the relevant authorities. Therefore, n_b and L_b are the only unknown parameters. It is then possible to rewrite (6) in order to have L_b described as a function of n_b :

$$L_b = \frac{1}{n_b} * (L_C + L_s) - L_s \quad (7)$$

However, L_b is now a decreasing function of n_b and, for it to stay positive, it is required to introduce a maximum value for n_b :

$$n_b \leq n_{sat} = 1 + \frac{L_C}{L_s} \quad (8)$$

This number indicates the highest possible number of virtual blocks before the circuit becomes saturated. The final step of this analytical description is the identification of the subarcs of f which will describe the movement of each virtual block. In other words, it is necessary to calculate the “phase shift” values which will satisfy the separation length between each block. Since all virtual blocks are arranged in a sequence with the same curvilinear length separating each other (assumption c), it is possible to decompose the integral in (3) into n_b equal integrals:

$$\frac{L_C}{n_b} = \int_{u_{k-1}}^{u_k} \sqrt{\left(\frac{dx}{du}\right)^2 + \left(\frac{dy}{du}\right)^2 + \left(\frac{dz}{du}\right)^2} du \quad (9)$$

With $2 \leq k \leq n_b + 1$, $u_1 = 0$, $u_{n_b+1} = u_p$

Fig. 2 depicts the relation described in (6) and (7). What can be put into perspective from it is the maximal number of blocks before saturation, which can be seen at the point where the length of the virtual blocks becomes null.

Using numerical tools, it is now possible to compute the position of each virtual block on the parametric arc.

B. Numerical Simulations

The previous section has laid down the required analytical background in order to successfully run numerical simulations. Furthermore, there have been no restrictions on the geometry of the parameterized arc so multiple configurations can be considered. This subsection will showcase four different simulations in order to illustrate the flexibility of the method. The

Fig. 2. Diagram representing the relation between the safety length, the number of blocks and the length of the virtual blocks, for a total arc length of $L_c = 10 \text{ km}$

chosen arcs have different specificities such as whether they are two or three dimensional, or whether they are built upon a single or a piecewise function.

The first simulation will revolve around the application of the carousel to a 2D ellipsoidal arc. The results for this arc have been depicted in Fig. 3.

Fig. 3. Two dimensional single arc

The next simulation will be built on the same basis as the previous one. The only difference stands in the addition of a sinusoidal vertical component. Results are depicted in Fig. 4.

The simulations represented in Fig. 3 and Fig. 4 depict continuous single arcs. The following two simulations shifts this approach towards piecewise built arcs. The arcs for those simulations will only be supposed continuous, the study regarding the impact of the differentiability of those arcs is out of the scope of this paper.

In other words, the following arc's analytical definition

Fig. 4. Three dimensional single arc

will change depending on the considered parameter-interval. Furthermore, these subarcs were defined in order to keep the continuity of the total arc. Fig. 5 depicts an example of such an arc.

Fig. 5. Two dimensional piecewise arc

In a similar way to the second simulation, the following one will add a vertical component to the previous arc. Furthermore, the vertical component will also be a piecewise function, as depicted in Fig. 6.

As the previous numerical simulations illustrate, there are no restrictions on the geometry of the parameterized arc. As such, it is safe to assume that the Carousel circuit can be designed in an unlimited amount of ways. Consequently, the Carousel circuit can be applied to any geometric configuration.

III. CASE STUDY

In order to help envisioning the possibilities such a method can offer, a case study has been designed. This study focuses on a potential operation consisting of regulating drone traffic

Fig. 6. Three dimensional piecewise arc

around a specific landing point. In this case, all virtual blocks are following a circular circuit around a runway, to which they are consecutively commanded to land on, using specified exit points and depending on their remaining battery.

The numerical simulation was run using a circular circuit with a radius of $R=2\text{km}$. The runway was designed as a rectangle whose vertical length is 0.5km and its horizontal length is 1km . Those two elements were arbitrarily centered on the point $(4\text{km}, 10\text{km})$. Furthermore, four exit points were arbitrarily chosen on the circuit, and their subsequent exit roads were computed using cubic interpolation. Fig. 7 depicts this considered geometry.

Fig. 7. Geometrical depiction of the circuit

Since the circuit is circular, the total arc length can be written as $L_C=2\pi R$. Furthermore, the separation length was taken as being twice the length of a virtual block: $L_s=2L_b$. Consequently, a new expression for the number of blocks can be obtained:

$$n_b = \frac{2}{3} \left(1 + \frac{\pi R}{L_b} \right) \quad (10)$$

Expression that satisfies (8).

Finally, the length of the virtual blocks was chosen as 0.5km , which leads to the total number of blocks: $n_b=9$. Fig. 8 represents the repartition of the virtual blocks alongside the circuit.

Fig. 8. Initial configuration with all virtual blocks (blue squares) circulating along the circuit

So far, the simulations in the previous section only focused on the circulation of the blocks alongside the circuit. The next step of the simulation requires the virtual blocks to successively enter one of the exit roads, as well as land on the runway. Fig. 9 shows one of the blocks following one of the exit roads, while Fig. 10 shows that same block later on, when it is on the runway.

Fig. 9. Virtual blocks circulating alongside the circuit (blue squares) with one block on an exit road (green square)

Fig. 10. Virtual blocks circulating alongside the circuit (blue squares) with one block on the runway (green square)

Fig. 9 and Fig. 10 represent the landing procedure for the first block. Once its successful landing has been confirmed, the next chosen block engages the procedure. This process continues until all blocks have landed. Fig. 11 shows the circuit situation after five blocks have landed.

Fig. 11. Circuit situation after five virtual blocks have landed

At this point, the configuration of the circuit as well as the landing procedure has been programmed. In order to include the remaining battery expectancy for each drone, it is first necessary to introduce time into the simulation. Indeed, the parametric variable u , which was used in the analytical description, does not correspond to time. It is thus required to find the relation between the variable u , and the time t . As the method was originally defined earlier, the key notion to do so lies in the constant speed the blocks are moving at. For this case study, the speed has been arbitrarily fixed at $v=40\text{km/h}$. Consequently, it is possible to obtain the time required for one block to travel along one circuit lap:

$$t_{\text{circuit}} = \frac{L_C}{v} = 18\text{min}51\text{s} \quad (11)$$

Since all blocks are moving at the same constant speed, it is also possible to fractionate (3) as follows:

$$\frac{t}{t_{\text{circuit}}} L_C = \int_0^{u_t} \sqrt{\left(\frac{dx}{du}\right)^2 + \left(\frac{dy}{du}\right)^2 + \left(\frac{dz}{du}\right)^2} du \quad (12)$$

With u_t , the parametric variable corresponding to time t .

Furthermore, the circuit being circular gives the relation between u_t and t :

$$u_t = 2\pi \frac{t}{t_{\text{circuit}}} \quad (13)$$

It is worthwhile to note that, the circuit being periodic, there is a need to adjust (13) by replacing t by $t - k \cdot t_{\text{circuit}}$ with k the number of circuit laps the virtual block has already traveled.

Now that the configuration has been switched to a time perspective, the remaining battery expectancy can finally be implemented. Additionally, the time each drone spends on the runway will also be introduced. Indeed, each block will spend a different time on the runway, which will be randomly chosen between 2 and 20 minutes for this case study. The same goes for the remaining battery expectancy, which will also be allocated by randomly selecting a remaining time between 1 and 5 hours. The program will then automatically select the virtual block with the least remaining battery to proceed on the landing procedure. Table. I summarizes the results obtained for one simulation in this configuration.

TABLE I
TABLE I. INITIAL AND RESULTING TIME DATA OF THE SIMULATION

Index number of virtual block	Result time ^a			
	Remaining battery expectancy at starting time	Remaining battery expectancy after landing	Time spent on the runway	Elapsed time between starting time and landing
1	04:13:24	02:30:51	15:49	01:42:39
2	02:18:21	01:39:42	05:53	00:38:41
3	03:11:59	02:05:37	02:37	01:06:25
4	02:33:19	01:36:00	09:51	00:57:22
5	04:35:14	02:10:28	18:52	02:24:55
6	03:42:16	02:23:29	06:43	01:18:53
7	04:18:49	02:18:22	12:15	02:00:35
8	01:26:25	01:11:54	08:28	00:14:32
9	02:07:01	01:44:12	02:29	00:22:50

^aAll times are in hh:mm:ss format.

It can be concluded from Table. I that all drones have successfully landed before any of them ran out of battery.

IV. DISCUSSION AND FURTHER CONSIDERATIONS

The Carousel method has been shown to have an advantageous geometric flexibility, and the case study displayed its application to a particular operational scenario. While the

method is still in its early stage, it is already possible to envision how it can be applied in the future. On the other hand, this paper only introduces the first step of the method. Indeed, there is still a vast amount of enhancements which can be applied to it and will thus need further research. Whether it is regarding the Carousel method itself or looking forward into an operational point of view, there are several ways to ameliorate and optimize it.

For example, the case study showed a sequence of virtual blocks leaving the circuit one by one. Consequently, gaps are created between the blocks remaining on the circuit (see Fig. 9-10-11), thus becoming empty slots which could be further exploited. The first possibility would be to immediately fill those slots with new users, while another possibility could be to automatically adapt the rest of the traffic by modifying the speed and augmenting the safety length, thus creating a new overall configuration, with a fewer amount of blocks.

Additionally, another point to consider would be regarding the dimensions of the blocks. Indeed, it might be necessary to distinguish blocks of different dimensions depending on the size, weight and general speed of the vehicle.

Furthermore, while the last two points consider a dynamic behavior of both the blocks and the traffic flow, another aspect to take into account would be regarding the dynamic behavior of the circuit itself. Indeed, machine learning methods could be applied to the circuit in order to create a dynamic in which the circuit will automatically adapt itself to the evolution of the traffic. This could lead to an intelligent circuit which can predict and respond to disruptions such as weather turbulences or congested traffic.

Regarding the circuit, the simulations so far have only displayed the capabilities of the Carousel method for a single, unidirectional track. Another approach would be to look at how the Carousel would behave if there were multiple parallel tracks, going in both directions. Such an approach would require higher levels of connectivity, which could ultimately lead to more interactions between different circuits, thus potentially creating a whole aerial, unmanned network.

Those are just a few examples among the myriad of possible considerations. While the Carousel method displays encouraging perspectives for the future of unmanned aircraft systems traffic management, it is still in a conceptual stage and will require further attention before its development can move onto the next stage.

REFERENCES

- [1] SESAR JU – U-space Blueprint
- [2] European ATM Master Plan – Roadmap for the safe integration of drones into all classes of airspace.
- [3] Koyuncu, Emre, and Gokhan Inalhan. "A probabilistic B-spline motion planning algorithm for unmanned helicopters flying in dense 3D environments." In 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 815-821. IEEE, 2008
- [4] Liu, Yang, Xuejun Zhang, Xiangmin Guan, and Daniel Delahaye. "Adaptive sensitivity decision based path planning algorithm for unmanned aerial vehicle with improved particle swarm optimization." *Aerospace Science and Technology* 58 (2016): 92-102.
- [5] Jenie, Yazdi I., Erik-Jan van Kampen, Joost Ellerbroek, and Jacco M. Hoekstra. "Taxonomy of conflict detection and resolution approaches for unmanned aerial vehicle in an integrated airspace." *IEEE Transactions on Intelligent Transportation Systems* 18, no. 3 (2016): 558-567.
- [6] Weisstein, Eric W. "Arc Length." From MathWorld—A Wolfram Web Resource. <http://mathworld.wolfram.com/ArcLength.html>