USING CELL-DEVS FOR PROTOTYPING UNMANNED AIRCRAFT SYSTEM TRAFFIC SIMULATION

Ifeoluwa Oyelowo
Dept. of Systems & Computer Engineering
Carleton University
1125 Colonel by Drive,
Ottawa, K1S 5B6, ON Canada
ifeoyelowo@cmail.carleton.ca

Bruno Artacho
Faculty of Engineering and Applied Science
Memorial University of Newfoundland
230 Elizabeth Avenue,
St. John's, A1C 5S7, NL Canada
bmartacho@mun.ca

Siu O'Young
Faculty of Engineering and Applied Science
Memorial University of Newfoundland
230 Elizabeth Avenue,
St. John's, A1C 5S7, NL Canada
oyoung@mun.ca

Gabriel A. Wainer
Dept. of Systems & Computer Engineering
Carleton University
1125 Colonel by Drive,
Ottawa, K1S 5B6, ON Canada
gwainer@sce.carleton.ca

ABSTRACT

The use of Unmanned Aerial Systems (UASs) is expanding speedily. This results in a need to integrate UAS traffic into non-segregated airspace. However, this integration introduces risks of mid-air collisions between UASs and manned aircraft (MA) in the airspace. To deal with these issues, we present two models that work together to assess the risk of a mid-air collision between a UAS and another manned aircraft operating in Canada's Northern airspace. The first model represents a part of Canada's Northern airspace and the aircraft and UAS traffic in it. The second model is an Uncorrelated Encounter Model (UEM) which determines whether or not a mid-air collision has occurred between a UAS and an aircraft. These two models are integrated to form a UAS-UEM model which determines the number of mid-air collisions in several UAS and aircraft flight simulations. Our results show a low probability for a mid-air collision between a UAS and an aircraft in the airspace region of interest.

Keywords: Unmanned Aerial System (UAS), mid-air collision (MAC), Cell-DEVS

1 INTRODUCTION

Unmanned Aerial Systems (UASs) were originally used for operations that were not suitable for human beings, such as military operations. However, today they are being used for several purposes such as product delivery, aerial photography, surveillance, etc. The use of UASs has expanded drastically over the years resulting in a need to integrate UAS traffic into non-segregated airspace. This integration will allow the public to fully benefit from the potential of unmanned aerial systems. This introduces the risk of mid-air collisions (MACs) between UASs and manned aircraft in the airspace, as UASs have no onboard pilots to detect and avoid conflicting traffic. Transport Canada requires that "UAS operations maintain an equivalent level of safety to manned aircraft operations" (Transport Canada 2017). But the existing methods for assessing risk is "subjective, lacks scientific rigor and produces results that can vary significantly between inspectors" (Transport Canada 2017).

To deal with this issue, we present a UAS traffic model that assesses the probability of "loss of separation" (LOS) and/or mid-air collision (MAC) between a UAS and manned aircraft operating in Canada's Northern airspace. The model integrates a discrete-event component that imitates Canada's Northern airspace based on actual flight patterns, and an Uncorrelated Encounter Model (UEM) which

determines the near mid-air collisions rate (NMAC rate) and uses this rate to estimate the rate of actual mid-air collisions.

2 BACKGROUND

The integration of UAS into the airspace shares the same common concept: first, the real necessity of implementing UAS in the non-segregated airspace with its involved risks; secondly, to establish regulations to avoid undesired hazards. The International Civil Aviation Organization (ICAO) establishes bases for subsequent regulations at Circular 328, "Unmanned Aircraft Systems (UAS)" (International Civil Aviation Organization 2011) in which it specifies as the main goal for aviation regulatory contributions as obtaining and maintaining the highest possible and uniform level of safety. It also states that when considering UAS, the goal is maintaining and ensuring the current level of safety, including other aircraft and safety of persons and properties on the ground. The Federal Aviation Administration (FAA) includes UAS integration in the National Airspace (NAS) as part of its recent publications (Federal Aviation Administration 2013) and highlights the non-damaging interaction of UAS with the already existent elements in the airspace.

Countries have been engaged to develop regulatory measurements that allow a certain level of safety in the airspace. For example, Civil Aviation Safety Authority (CASA) has an ongoing project which will introduce a regulation and guidelines to the industry in the operation of UAS in the Australian Airspace (Civil Aviation Safety Authority 2017). In more recent years, academic studies were conducted to determine risk analysis of flying UAS over inhabited areas in Australia (Clothier et al. 2007). The level of safety is also mentioned by EUROCONTROL in its latest document related to UAS, in this case referenced as Remotely Piloted Aircraft (RPA) (EUROCONTROL 2012). The UAS integration with the European Airspace was successfully conducted in the German Air Traffic Control (ATC), as described in (Ruff-Stahl et al. 2016). During the operations at the German Air Space, the use of hardware applications that increased safety levels is relevant, such as Traffic Collision Avoidance System (TCAS) and lighting/marking on the aircraft, as well as the complete immersion of UAS into the ATC.

Given these regulatory entities directives and the risks derived from the UAS integration in the airspace, we need to perform a risk assessment to successfully develop a Detect and Avoid (DAA) method that permits the UAS implementation in the airspace, without the requirement of a ground pilot. Such evaluation of safety levels is performed by the analysis of different metrics, such as the Mid-Air Collision (MAC) and the Near Mid-Air Collision (NMAC) rates. The risk assessment of the UAS introduction into the airspace can be successfully evaluated by the determination of those metrics with the incorporation of UAS. The MAC and NMAC rates provide essential insight of the safety associated with air traffic operations. Both MAC and NMAC rates have been progressively decreasing over the past decades worldwide. This fact is credited to the modernization of radars and systems adopted for ATC.

As a guideline for the development of risk assessment methods in UASs, (Walker and Clothier 2015) present steps that lead to the system development and evaluation. The risk identification stage establishes origin and cause for failures. Likewise, there are several established techniques by (Janic and Netjasov 2008) for risk identification:

- Event Tree Analysis (ETA) isolates hazards to evaluate the outcome of sequential failures. It models the scenario by analyzing the success or failure in a singular initial trigger event, following paths according to probabilities and outcomes (Melnyk et al. 2014).
- Fault Tree Analysis (FTA) combines events that lead to a hazard. It is often applied combined with ETA, as seen in (Stroeve, Blom and Bakker 2009), aiming to simulate incursions on airports runways.
- TOPAZ analyzes the scenario combined with Monte Carlo Simulation.

The main function of an Unmanned Encounter Model (UEM), is to generate scenarios between the UAS and the intruder aircraft with a potential for a hazardous event, based on observed real flight data (Kochenderfer et al. 2008). The UEM we use is a modification of (Weibel, Edwards and Fernandes 2009), which was built by request of the FAA and US Departments of Defense (DoD) and Homeland Security (DHS). It considers 130 radars and almost a year of data in the NAS. Figure 1 presents the radar data acquired in flight hours divided into 1/6 of a degree by 1/6 of degree cells.

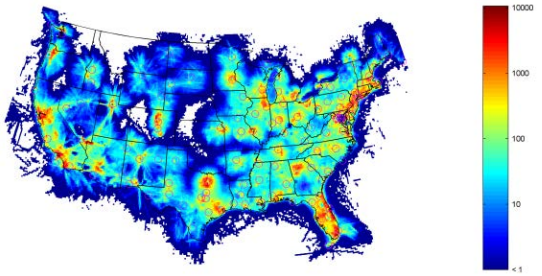

Figure 1: Acquired data in flight hours for the encounter model in NAS (Kochenderfer et al. 2008)

The model is currently adopted by the Radio Technical Commission for Aeronautics (RTCA-SC228) for its risk analysis and tests, and is, therefore, a method with proven acceptance and efficiency. As presented in (Kochenderfer et al. 2008), this UEM takes into consideration altitude class and layer, airspeed, acceleration, turn rate, and vertical rate as variables for the model. With the initial state and transitions dynamically modeled as a stochastic Markov model, tracks are created following aircraft tracks observed during actual flights through NAS. The UEM simulates encounters including aircraft flying under Visual Flight Rules (VFR), where the manned aircraft (MA) does not see the UAS artificially inserted into the traffic pattern. The model also enables the capability for collision avoidance algorithms implementation.

In (Jackson and Boskovic 2012), there are two methods of assessing the collision risk: The Maximum Reachable Sets (MRS), which relies on the determination of points where the UAS can reach within a fixed turn rate and velocity over a short period of time. The second method is the Probabilistic Reachable Sets (PRS). It relies on typical flight dynamics observation to set boundaries according to observed probabilities of trajectories. The PRS approach determines three-dimensional boundaries considering probabilities for different bearing angles and altitude differences.

We will use a UAS model (described later) to simulate the airspace at a high-level while considering both the intruder's and UAS's flight path. The models of interest should include both continuous (mid-air collision models) and discrete-event (origin/destination) model components, combined with spatial models. For instance, the behavior governing the mid-air collision model is described with differential equations (continuous modeling technique), while the Radar models might be better modeled using a discrete-event formalism. In the last few years, different approaches have been proposed to simulate continuous systems under the discrete-event paradigm. Spatial notions can provide extra facilities for understanding and visualizing the resulting simulation. We propose to address these issues by building a discrete-event model based on the DEVS formalism (Zeigler, Kim and Praehofer 2000), a framework for the modeling and simulation (M&S) of discrete-event systems. DEVS provides an abstract approach to modeling and simulation by separating the modeling from the simulation aspects and hence facilitating the model usability and composability. Furthermore, DEVS permits defining complex behavior as a form of emergent behavior produced by the interactions among simpler subsystems. Any subsystem belongs to a hierarchical structure, being at the same time a composition of finer grain subsystems and a component of larger systems. DEVS also provides a self-contained and mathematically sound specification of models with strict separation from any simulation technology or computer language. This feature allows the model continuity of the studied environmental systems, fostering their reuse and incremental refinement by different research teams independently from the simulation technologies that might change across

teams and time. The basic building block of any DEVS model is the *atomic* model, which can be connected to other atomic models to form what is called a *coupled* model.

We will investigate the possibility of describing the spatial interaction of the discrete-event models using the Cell-DEVS formalism. Cell-DEVS allows modeling discrete-event applications with spatial components using a continuous time base, providing instantaneous events that can occur asynchronously at unpredictable times. In addition, these models can be easily composed of models defined in other formalisms. Cell-DEVS models are defined as a space composed of individual cells that can be lately coupled to form a complete cell space. Each cell is a continuous time model, defined by very simple rules and a few parameters. Complex timing definition is overruled due to the use of different delay functions. A sample Cell-DEVS coupled model is presented in Figure 2a.

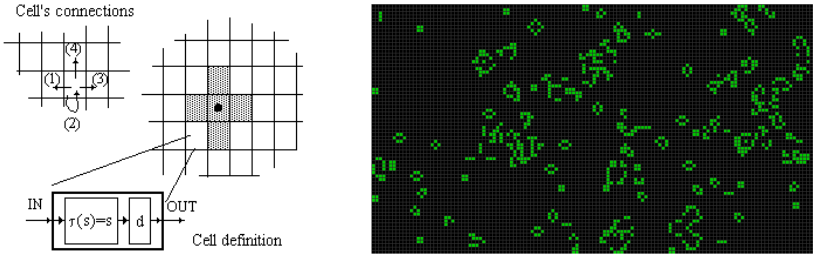

Figure 2: a) Cell-DEVS coupled model. b) Processing Implementation of Conway's Game of Life. (Soler-Adillon 2012)

CD++ (Wainer 2002) is an M&S environment developed in C++ following the formal specifications of DEVS and Cell-DEVS. It is used to build and execute DEVS and Cell-DEVS models. DEVS Atomic models are programmed in C++ and incorporated into the CD++ class hierarchy. Once an atomic model is defined, it can be combined with others into a multi-component model using a specification language specially defined for this purpose. Interfacing the models with others defined in external functions is simple (Lopez and Wainer 2004).

Our Cell-DEVS prototype was used to establish the basic model's rules that were later modified and improved on in the Processing tool (Fry and Reas 2012), an open source computer programming language and integrated development environment (IDE) designed to "facilitate computer vision, data visualization, music composition, networking, 3D file exporting, and programming electronics." (Fry and Reas 2012). Figure 2b shows an example of a Processing program (Soler-Adillon 2012) which is a Processing implementation of John Conway's famous Game of Life. In this work, we made use of the Processing tool because it has a large collection of libraries which we found useful for the development of the UAS model. In addition, we could visualize the results of the model using the tool.

3 UAS AND UEM MODELS

The UAS model is a discrete-event model which imitates to an extent a segment of Canada's Northern airspace and the flight traffic in that region of the airspace. The model uses flight data from Canada's Northern airspace provided by Nav Canada as a guide to model and simulate realistic aircraft flight traffic in the airspace. To generate several flight paths different from the ones provided by Nav Canada, the model uses the provided data as a base. The UAS model was implemented with two different methodologies discussed in this section. The first implementation involves the Cell-DEVS formalism and software, while the second implementation involves the Processing software. The Uncorrelated Encounter Model (UEM) determines the rate of near mid-air collisions (NMAC rate) and uses this rate to estimate the rate of actual mid-air collisions. The UAS model and the UEM model were integrated to conduct advanced experimentation.

3.1 Cell-DEVS Implementation of the UAS model

In the Cell-DEVS implementation of the UAS model, a cell space was used to represent Canada's Northern airspace. We used a two-dimensional 50x50 grid using Moore's Neighborhood of 9 cells. Each cell uses three state variables: *position, path, and sourceDest.* Each cell also uses the neighbor ports *positionport, pathport,* and *sourceDestport* to transmit the values of a cell to its neighborhood.

- Position: We use the value 100 to indicate that the cell contains a UAS; if position=101 the UAS is on a predefined flight path and can continue moving on that path. If position=102 for a cell, the UAS in that cell is not on a predefined flight path. A value between 1-8 indicates the direction of the next cell in the UAS's path. Position = 50 for a cell indicates that the UAS has visited that cell in the past. Position = 150 for a cell means that the UAS has reached its destination which is in that cell. The default value is 0: the cell does not have a UAS or it has not been previously occupied.

- **Path:** This variable can have values 0 to ∞ to mark the path taken by the UAS.

- **SourceDest:** We use values 0, 200 or 201, to indicate if the cell is a source (200), a destination (201) or none (0).

The Cell-DEVS definition of the model is based on a set of rules summarized here.

A cell that has a UAS (position = 100) will change to 101 if the cell has sourceDest = 200 (the UAS is at the start of a path). If a cell contains a UAS but the cell does not start a path **(**sourceDest ≠ 200), then position = 102 for that cell. If a cell has position = 101, this means that the UAS is on a path going to its destination. And if a cell has position = 102, this means that it is a UAS not on a path and will not be moving.

Figure 3 illustrates the above rules. In Figure 3a, we can see that the cell space has two cells with position = 100 which means that these two cells contain UASs. After one invocation of the rules, the cell space updates (shown in Figure 3b) according to the rules described above. Note that the cell spaces (5x5 grids) shown in Figure 3 and Figure 4 are used only to better explain the model's rules.

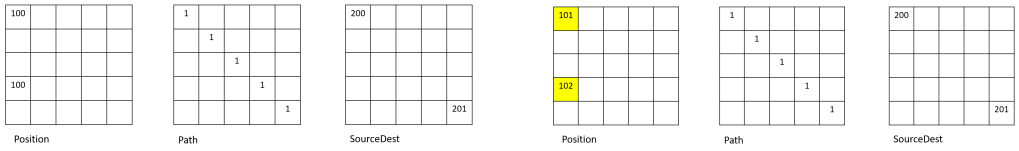

Figure 3: a) Initial Values of the Cell Space. b) Cell Space one time step after Figure 3a.

Figure 4 shows the cell space after executing the rules after the configuration seen in Figure 3b. The cell with position = 101 checks its neighborhood to see which of them has the same value for its path variable (path = 1) as itself. In this case, it finds the cell that meets the condition, and that cell is given a value for its position variable. We can see (in Figure 4a) that the cell with position = 6 has a value of 1 for its path variable. Therefore, that will be the next location for the UAS (in Figure 4b, the cell that had position = 6 now has position = 101). We can also see that the previous cell location of the UAS now has position = 50. This lets us track the cells that the UAS has visited.

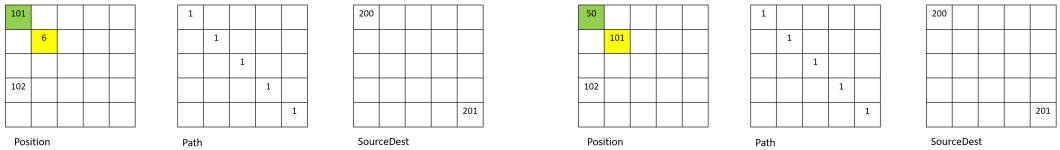

Figure 4: a) Cell Space one time step after Figure 3b. b) Cell Space instantaneously after Figure 4a.

These rules and cell space were implemented in CD++, and numerous simulations were executed to learn the basic behavior in the cell space. The Cell-DEVS UAS model has a scope that is limited to multiple UASs travelling simultaneously on separate paths (not overlapping paths). However, the intruder aircraft was never introduced into the model. The UAS's flight paths tested were randomly generated. We used this Cell-DEVS implementation as a proof-of-concept model, showing how to simulate the UAS behavior.

3.2 Processing Implementation of the UAS Model

The Processing implementation of the UAS model is a more elaborate model compared to the Cell-DEVS version of the UAS model and it is closer to the requirements specified in (Transport Canada 2017). The Processing UAS model simulates the travel paths of a UAS and an aircraft and checks if there is a near mid-air collision (NMAC) between them. In the Processing UAS model, the intruder is introduced into the model. However, the model's scope was limited to consider only one aircraft and UAS pair for each simulation run to reduce the complexity of the model. The model was divided into three parts: The Main class, the Aircraft class, and the UAS class, discussed in the sections below.

- UAS Class: The UAS model assumes that 1 pixel (in the sketch's display window) represents 1 km in the airspace and this influenced the units of our state variables. Each UAS has an **ID** (int), its **position** (of type *PVector* holding x and y coordinates of the position of the UAS in km), its **velocity** (another *PVector* with the velocity of the UAS in km/s), the **initialPosition** and **destinationPosition** (two *PVectors* with the x and y coordinates of the point at which the UAS will begin/end its path), the **maxSpeed (**float, km/s**),** **altitude** (int, with the altitude of the UAS in ft). PVector is a 2/3D Euclidean vector (i.e., a variable with both magnitude and direction). In the examples we will execute, maxSpeed = 0.051 km/s (or 100 knots), based on the UAS requirements; likewise, altitude = 10000 ft (the actual UAS will operate at a low altitude of 2000-5000 ft AGL, or a medium altitude of 5000-20000 ft MSL according to (Transport Canada 2017)).

The velocity of a UAS is calculated as the difference between the destinationPosition and the current position of the UAS but limited to the maxSpeed state variable. The position is calculated as the current position added to the velocity multiplied by simTimeDelta, a variable that represents the discrete-time step in the model.

- **Aircraft Class:** It ensures that the modelled aircraft has realistic features. Each instance of this class includes an **ID** (int), its **position** (of type *PVector,* holding x and y coordinates of the position of the aircraft in km), its **velocity** (another *PVector* with the velocity of the aircraft in km/s), the **maxSpeed (***float*, km/s**),** and **altitude** (*int*, with the altitude of the aircraft in ft). It also includes **flightPath** (an array of *PVector* with all x, y coordinates for a single aircraft flight path obtained from real data in the simulation area); **randflightPath** (an array of *PVector* with modifications to the original flightPath, to generate randomness to the source data), and **flightTime (**an array of *int* that includes the real *time* for each of the data points in the flightpath). The values of flightPath, randflightPath, flightTime and Altitude arrays are from a Canadian flight database provided by Nav Canada, which contains *Time, X, Y and Altitude* (*Time* is the real time, in seconds in which each path data point was reached). Processing and CD++ use a coordinate origin at the top left; however, the source data's coordinate origin is at the bottom left. We convert the origins and adjust their size (the data is in meters, and in our model, one pixel represents 1 km). A random deviation within a specified range ([-11,11] km) is combined with the original flightPath, to form the randflightPath array that the aircraft use to simulate travel. Then,

$$velocity = \frac{position - randflightPath[timeIndex - 1]}{time - flightTime[timeIndex - 1]}, limited\ to\ maxSpeed, \text{ where timeIndex} \neq 0$$

where time = flightTime[timeIndex] and position = randflightPath[timeIndex].

- **Main Class:** the UAS and Aircraft objects then travel on their respective flight paths in the airspace and we check to see if there will be a collision. To do so, we use **simTime** to store the simulation time,

collisionRadius (float, in km) to specify the maximum distance between the aircraft and UAS at which we consider a near mid-air collision (NMAC) to have occurred, **backgroundMap (**type *PImage*) to hold a satellite image of the airspace of interest, **aircraft** and **UAS,** instances of the class Aircraft and UAS.

In our case, collisionRadius = 9.26 km: if the aircraft and the UAS are within this distance and their altitudes are within 1000 feet, we consider this to be an NMAC. The visualization includes a 2D environment with a size of 1000 x 1000 pixels, using a framerate of 120/s.

We also collect information about different metrics for statistical purposes: **TotalSimtime (s)** as **simTime** is reset for each new pair of Aircraft and UAS objects, **NumberofAircrafts** (represents how many aircraft and UAS pairs are simulated, based on the source file with the flight paths data), **Col_det_table** (stores information about the UAS and aircraft when a collision occurs, in particular the position and velocity information of both the aircraft and UAS) and **Collided_ID,** the ID of the aircraft that just collided.

3.3 The Uncorrelated Encounter Model Implementation

The model is built in MATLAB, including 60 seconds of data acquired from MIT LL UEM. The system uses the initial position and heading angle, as well as its variation for every second, totaling a value superior to the required 2τ + 15 seconds proposed by (Baillie et al. 2016). The required distance for collision avoidance was calculated taking into consideration different velocities for both the UAS and intruder aircraft in all possible bearing angles and assuming the worst-case scenario for its flight, that is direct collision route for the aircraft. Consequently, the simulations using the UEM were conducted within an Encounter Cylinder of radius 3 km and a height of 200 ft.

The dynamic system for tracking is defined using the relative position of the intruder in the own-ship aircraft (UAS) frame. Therefore, to derive the equations for position, velocity, and relative heading angle, it is necessary to consider the rotation of both aircraft, resulting in the following equations for longitude, latitude and heading angle, respectively as defined in (1):

$$\begin{bmatrix} \dot{p_x} \\ \dot{p_y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v_{intruder} \cdot \cos\theta - v_{UAS} + \omega_{UAS} \cdot p_y \\ v_{intruder} \cdot \sin\theta - \omega_{UAS} \cdot p_x \\ \omega_{intruder} - \omega_{UAS} \end{bmatrix} \quad (1) \qquad y = \begin{bmatrix} r \\ \beta \end{bmatrix} = \begin{bmatrix} \sqrt{p_x^2 + p_y^2} \\ \tan^{-1}(\frac{p_y}{p_x}) \end{bmatrix} \quad (2)$$

Here, p_x and p_y represent the intruder relative positions in coordinates x (longitude) and y (latitude), respectively, and θ represents the relative heading angle for the intruder. The velocities (v) and angular velocities (ω) are also considered to predict flight positions. Also, to measure the distance between the aircraft and the bearing angle, the system measurements are defined as in (2), where **r** represents the range between aircraft and β is the bearing angle between the UAS and intruder. Figure 5 shows the comparison of states performed at every iteration between aircraft, checking for relative distance and potential NMAC, and evaluating if the collision risk value is acceptable. The bearing angle and velocities are used to calculate the specific value of τ for that scenario, and the time for avoidance maneuver is calculated.

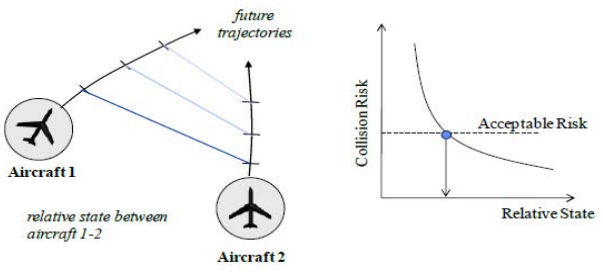

Figure 5: State and risk for the simulation (Weibel, Edwards and Fernandes 2009)

To evaluate different scenarios, we run simulations with different ranges, using $2\tau + 15$ seconds for collision avoidance. We compute the NMAC rate with and without radar detection and avoidance considering the required distance for avoidance. As presented on (Kochenderfer et al. 2008), the UEM takes into consideration the altitude class and layer, airspeed, acceleration, turn rate, and vertical rate. Tracks are created following aircraft tracks observed during actual flights through the NAS. To calculate the probability of NMAC, we consider the traffic density of the area being analyzed (Kochenderfer et al. 2008). Definitions of point flight dynamics are used to simulate flight paths for both the UAS and intruder, following the procedure described in (Jackson and Boskovic 2012) and (Sahawneh and Beard 2014).

3.4 Integration of the UAS and UEM models

The UAS-UEM model works such that the UAS model is run first and then the UEM model computes the final results. The UAS model oversees the high-level flight traffic simulation of the intruder and the UEM model can estimate if there will be a mid-air collision when the two are within close range. The UAS model is run with several aircraft and UAS travel paths (about 328 different flight paths based on Nav Canada traffic flight information). The UAS model was implemented using the Processing software. The three parts of the model (The Main class, the Aircraft class and the UAS class) are exactly as previously explained.

4 CASE STUDY

In this section, we present the results of a case study that shows the simulation of the UAS model and the UEM. In addition, we provide the results of the integrated model.

As discussed earlier, we built a Cell-DEVS version of the UAS model to quickly prototype the basic behavior of the model. Figure 6a shows the initial condition: the left part of the figure shows the initial value of the *position* state variable in the entire cell space, the middle section shows the initial values for the *path* state variable and the right side shows the initial values for the *sourceDest* state variable.

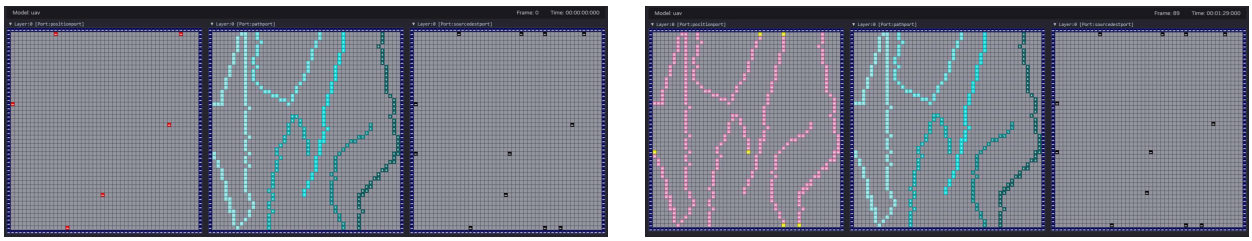

Figure 6: a) Initial Values of the Cell Space b) Cell Space at the end of the Simulation

Since there are six distinct paths and six UASs in this model, as expected, at the end of the simulation, each of the UASs has travelled on its own path and reached its destination. The cell space at the end of the simulation is shown in Figure 6b. We can see that each UAS (shown in the *position* plane) follows its own path (as defined in the *path* state variable plane) until it has completed the path and reached its destination.

As discussed in section 3, we used these initial results to analyze the discrete-event implementation of the model using Cell-DEVS and then built an improved version using the Processing programming language, as described in Section 3. Using the new simulation engine, we conducted numerous case studies. The one presented used the following initial values:

- **NumberofAircraft** = 328 (the total number of unique flight path data simulated).

- **SimTotal** = 1000 (we run 1000 iterations of random simulations chosen between the 328 different flight paths available in the area; as discussed earlier, the paths are basic guidelines and waypoints that are modified using random modifications to the path; no aircraft or UAS flight path is repeated).

- We collect information in **Col_det_table**, including *FlightID* (the ID of the aircraft that collided), *Xuas* and *Yuas* (the x and y coordinates, in meters, of the position of the UAS when the collision occurred), *V_Xuas* and *V_Yuas* (the x and y coordinates, in m/s of the velocity of the UAS when the collision occurred), *Xint* and *Yint* (the x and y coordinates, in m, of the position of the aircraft or intruder when the collision occurred), *V_Xint* and *V_Yint* (the corresponding velocity, in m/s).

  If there has been a collision, the required information is stored in the Col_det_table table. Only one set of collision information is stored per simulation run. When either the aircraft or UAS reaches its path end or destination, a different pair of Aircraft and UAS objects are created and we start a new run.

Figure 7 shows a snapshot of the simulation in which an aircraft is travelling on its flight path, and a UAS is travelling in the North West (NW) section (yellow dot). A visualization can be found at http://bit.ly/2DRPx3e.

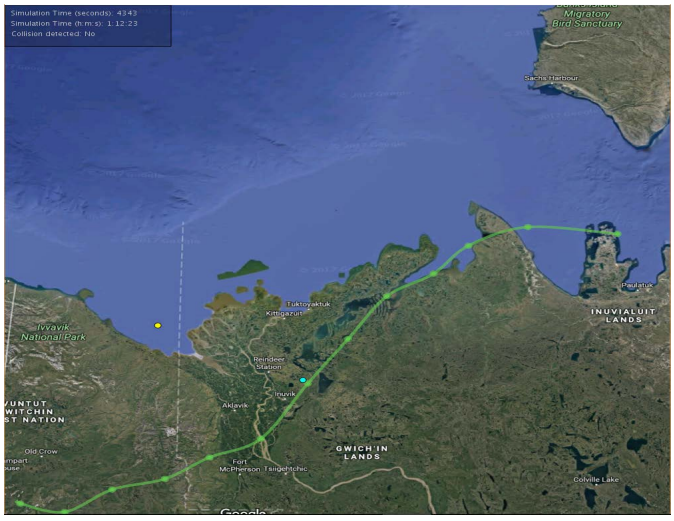

Figure 7: Snapshot of the Display Window taken while the UAS model was running

The green dots show the exact flight path data points read from the source data provided by Nav Canada for real flights. The light blue circle represents an aircraft travelling along the flight path. The yellow circle in the NW of the figure represents a UAS whose travel path is a randomly generated straight line. Both the UAS and the aircraft have a collision region (represented as shadows or transparent circles around both the UAS and aircraft) around them. These 2 circles touch when the NMAC condition occurs. The information box at the top left of the screen shows the simulation time (hh:mm:ss) and a message specifying if a collision has been detected or not.

The UAS should fulfill the requirement to not increase the NMAC rate in the airspace in which it is inserted. Also, the risk analysis for a UAS should consider the potential damage to other aircraft and hazards posed to population and constructions on the ground. In a complete UAS system, in which considering an implemented detect and avoid (DAA) system, a fault tree is defined analogously to (Stevenson, O'Young and Rolland 2015) by the equation:

$$P_{NMAC} = P_{SEP\ LOSS} \cdot P_{UAS\ FAIL} \cdot P_{MA\ FAIL}$$

Where P_{NMAC} represents the probability of NMAC posteriorly calculated, $P_{SEP\ LOSS}$ is the probability of separation loss occurring, $P_{UAS\ FAIL}$ is the probability of the UAS system failing to detect or avoid the intruder, and $P_{MA\ FAIL}$ is the probability that the intruder manned aircraft does not detect or avoid the UAS. If there is no implementation of DAA applications for the intruder, $P_{MA\ FAIL}$ is considered to have a unit value. From the Fault Tree for this system, the DAA failure component is equivalent to $P_{UAS\ FAIL}$, the probability that the intruder will be entering the Encounter Cylinder is $P_{SEP\ LOSS}$, whilst the NMAC rate is P_{NMAC}.

It is important to observe that the probabilities of failure described above are not independent. That is, components that might lead to a collision in the Fault Tree possess dependency, resulting in a system which is too complex to be determined without simulation.

When accounting for risk, as pointed out in (Park, Lee and Mueller 2014), the bearing angle of the intruder is an essential aspect to be observed. This allows efforts to be focused on detection and avoidance of collisions in specific bearing angle sections. As an initial analysis, intruders in the simulation were divided into different altitude levels: From 3,000 ft.-5,000 ft., and above 5,000 ft. In both scenarios, the intruder is simulated with its origin in the Encounter Cylinder previously defined. As expected, there is a higher concentration of NMAC for intruders with a frontal bearing angle, due to its higher relative velocity and traffic pattern. In addition, the results present a slightly lower number of NMAC for flights in the higher altitude layer.

In order to estimate the value of encounters per hour in a determined area, it is necessary to take into account the flight density of that particular airspace. As defined in (Kochenderfer et al. 2008), the NMAC rate, $\lambda_{NMAC,}$ is determined by:

$$\lambda_{NMAC} = P(NMAC|ENC) \cdot \lambda_{ENC}$$

Where P(NMAC|ENC) is the percentage of NMAC given that the intruder is within the Encounter Cylinder previously calculated, and λ_{ENC} is the rate of intruders penetrating the volume swept by the UAS multiplied by the constant air traffic density of the specific area under analysis, being posteriorly analyzed in further detail. Figure 8 demonstrates the distribution of NMAC above 5,000 ft. with the UAS flying without any detection or collision avoidance implementation.

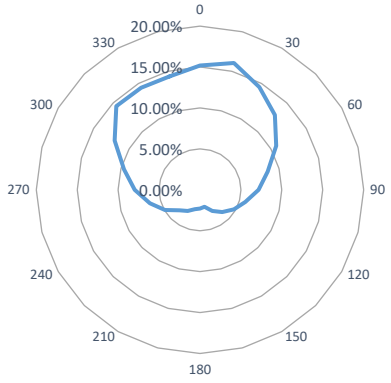

Figure 8: Intruder bearing angle distribution

Considering all bearing angles for flights inside the Encounter Cylinder, 0.235% of the total number of flights simulated in the lower altitude layer and 0.229% of flights simulated in the higher altitude layer are NMAC trajectories. From the collision trajectories, a total of 57.78% are concentrated on head-on collision bearing angles, while only 11.98% are originated from the tail-chase scenario. In the following equation, the rate of intruders penetrating the Encounter Cylinder is determined by the aircraft density (ρ) for a specific airspace in aircraft/nm^3, multiplied by the Encounter Cylinder swept volume (V):

$$\lambda_{ENC} = \rho \cdot V$$

The method developed was applied for analysis of NMAC rates for Transport Canada in the Canadian Arctic. As a preliminary analysis following the Big Sky Theory for aircraft distribution (aircraft are equally distributed over the airspace in which the analysis is being conducted, with dynamics being only included for close proximity intruders (Stevenson, O'Young and Rolland 2015), it can be determined that, for a UAS flying at 100 kt., with a turn rate limited to 2G (12°/s), intruder aircraft velocities smaller or equal to 170 kt., and assuming that most of the intruder aircraft are at altitudes equal or lower than 18,000 ft., the NMAC rate is estimated as $3.52 \cdot 10^{-7}$ NMAC/h. The preliminary analysis conducted by the implementation of the developed model concludes that the Arctic surveillance operation presents a low risk and can be conducted without the requirement of implementing detect and avoid capabilities, consequently reducing the development and implementation costs associated with the project.

5 CONCLUSION

We discussed the application of simulation for building a UAS traffic model to assess the probability of mid-air collision between a UAS and another aircraft operating in Canada's Northern airspace. In this report, two implementations of a discrete-event UAS model were described. We also discussed the Uncorrelated Encounter Model (UEM), which computes the NMAC rate, and showed the simulation results of an integrated model. The simulation is integrated with a graphical interface using actual GIS data, and real world traffic information provided by Nav Canada. A Cell-DEVS implementation of the UAS model provided a basic prototype to learn how to build the application. The Processing implementation of the UAS model is an improvement to the Cell-DEVS version and it includes the intruder's traffic information and the altitude of both the UAS and aircraft. In the future, the Processing implementation of the UAS model will be updated to include certain aspects of the airspace such as different seasons, weather conditions, different airspace classes etc. Finally, more simulations need to be run on the integrated model and more realistic UAS flight path data need to be obtained.

REFERENCES

Baillie, S., W. Crowe, E. Edwards and K. Ellis, "Small Remotely Piloted Aircraft System (RPAS) Best Practices for BVLOS Operations," Unmanned Systems Canada Conference, Edmonton, 2016.

Civil Aviation Safety Authority (CASA), "Project OS 11/20 - Review of Regulations and Guidance Material relating to Unmanned Aircraft Systems (UAS)," [Online]. Available: https://www.casa.gov.au/standard-page/project-os-1120-review-regulations-and-guidance-material-relating-unmanned-aircraft. Accessed August 2017.

Clothier, R., R. Walker, N. Fulton and D. Campbell, "A Casualty Risk Analysis for Unmanned Aircraft Systems (UAS) Operations over Inhabited Areas," 2nd Australasian Unmanned Air Vehicles Conference, 2007.

EUROCONTROL, "EUROCONTROL Specifications for the use of Military Remotely Piloted Aircraft as Operational Air Traffic Outside Segregated Airspace," European Organization for the Safety of Air Navigation, no. First edition - 2013. U.S. Department of Transportation. Federal Aviation Administration, 2012.

Federal Aviation Administration (FAA), "Integration of Civil Unmanned Aircraft Systems (UAS) in the National Airspace System (NAS) Roadmap," 2013.

Fry, B., and C. Reas, "Processing: Overview," Processing, 2012. [Online]. Available: https://processing.org/overview/. [Accessed 12 December 2017].

International Civil Aviation Organization (ICAO), "Unmanned Aircraft Systems (UAS)," International Civil Aviation Organization, 2011.

Jackson, J. A., and J. D. Boskovic, "Application of Airspace Encounter Model for Prediction of Intruder Dynamics," in AIAA Modelling and Simulation Technologies Conference, 2012.

Janic, M., and F. Netjasov, "A Review of Research on Risk and Safety Modelling in Civil Aviation," Journal of Air Transport Management, pp. 213-220, 2008.

Kochenderfer, M. J., L. P. Espindle, J. K. Kuchar and J. D. Griffith, "A Comprehensive Aircraft Encounter Model of the National Airspace System," Lincoln Laboratory Journal, 2008.

Kochenderfer, M. J., J. K. Kuchar, L. P. Espindle and J. D. Griffith, "Uncorrelated Encounter Model of the National Airspace System," Massachusetts Institute of Technology Lincoln Laboratory, Lexington, 2008.

Kochenderfer, M. J., L. P. Espindle, J. K. Kuchar and J. D. Griffith, "A Bayesian Approach to Aircraft Encounter Modelling," in AIAA Guidance, Navigation and Control Conference and Exhibit, 2008.

Lopez, A., and G. Wainer, "Extending CD++ Specification Language for Cell-DEVS Model Definition," Dept. of Systems and Computer Engineering. Carleton University, Ottawa, ON, 2004.

Melnyk, R., D. Schrage, V. Volovoiand and H. Jimenez, "Sense and Avoid Requirements for Unmanned Aircraft Systems Using a Target Level of Safety Approach," Risk Analysis, p. 2014.

Park, C., S. M. Lee and E. R. Mueller, "Investigating Detect-and-Avoid Surveillance Performance for Unmanned Aircraft Systems," in 14th AIAA Aviation Technology, Integration and Operations Conference, 2014.

Ruff-Stahl, K. H. J., S. Esser, D. Farsch, T. Graner and R. Klein, "Not-So-Risky Business? Assessing the Risk of Integrating Large RPVs into the Current Air Traffic System," International Journal of Aviation, Aeronautics and Aerospace, 2016.

Sahawneh, L. H., and R. W. Beard, "A Probabilistic Framework for Unmanned Aircraft Systems Collision Detection and Risk Estimation," 53rd IEEE Conference on Decision and Control, 2014.

Soler-Adillon, J. "Processing: Game of Life Example," Processing, 2012. [Online]. Available: https://processing.org/examples/gameoflife.html. [Accessed 12 December 2017].

Stevenson, J. D., S. O'Young and L. Rolland, "Estimated levels of safety for small unmanned aerial vehicles and risk mitigation strategies," NRC Research Press, 2015.

Stroeve, S. H., H. A. P. Blom and G. J. Bakker, "Systemic accident risk assessment in air traffic by Monte Carlo Simulation," Safety Science, p. 238–249, 2009.

Transport Canada, "Statement of Work - Preliminary Unmanned Aircraft System (UAS) Traffic Simulation", Ottawa, 2017.

Wainer, G. "CD++: A Toolkit to Define Discrete-Event Models," Software, Practice and Experience, Wiley, vol. 32, no. 3, pp. 1261-1306, 2002.

Walker, R. A., and R. A. Clothier, "Safety Risk Management of Unmanned Aircraft Systems," in Handbook of Unmanned Aerial Vehicles, Springer, 2015.

Weibel, R. E., M. W. M. Edwards and C. S. Fernandes, "Establishing a Risk-Based Separation Standard for Unmanned Aircraft Self Separation," Ninth USA/Europe Air Traffic Management Research & Development Seminar, 2009.

Zeigler, B., T. G. Kim and H. Praehofer, "Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems," Academic Press, 2000.