Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL FINAL DE GRAU

TÍTOL DEL TFG: Simulation of the ground movements and conflict detection

TITULACIÓ: Grau en Enginyeria de Sistemes Aeroespacials

AUTOR: Samantha Mercader Gonzalez

DIRECTOR: Jordi Pons i Prats

CO-DIRECTOR: Jovana Kuljanin

DATA: 05 de Setembre, 2022

Títol: Simulation of the ground movements and conflict detection

Autor: Samantha Mercader Gonzalez

Director: Jordi Pons i Prats

Co-director: Jovana Kuljanin

Data:

Resum

Amb el continu creixement del transit aeri, els recursos disponibles en períodes de gran activitat no poden satisfer la demanda del trànsit aeri, el que genera molta congestió. Aquesta no només afecta en la reducció de la eficiència operativa, sinó que també fa que s'augmenti la càrrega de treball dels controladors aeris, conseqüentment, les possibilitats de risc de invasió en pista o col·lisió en carrer de rodatge augmenten. Per aquest motiu, és imprescindible disposar d'un bon mètode de detecció i resolució de possibles conflictes entre vehicles.

L'objectiu d'aquest projecte és comprendre tots els aspectes que afecten a la generació de possibles conflictes entre vehicles a la zona de rodatge de l'aeroport. Per aconseguir aquest propòsit, s'ha realitzat un estudi de les principals característiques de la zona de rodatge de l'aeroport: velocitats, prioritats, senyalització, entre altres. Per altre banda, s'ha analitzat diversos mètodes estudiats i/o implementats de detecció de conflictes i de resolució per tal de poder aprofundir-ne més en l'àmbit.

En la segona part d'aquest projecte s'ha creat una simplificació de detecció i de resolució de conflictes mitjançant Python, l'objectiu principal és poder posar en acció tots els aspectes estudiats de l'aeroport de la forma més realista possible. Un cop realitzat, es podran visualitzar les diferents trajectòries i els possibles conflictes generats.

Globalment, el projecte proporciona algunes pautes sobre temes de detecció i resolució de conflictes que es consideren rellevants per a futures investigacions en aquest àmbit. A més, s'esmenten els elements necessaris per la implementació en l'aeroport.

Title: Simulation of the ground movements and conflict detection

Author: Samantha Mercader Gonzalez

Director: Jordi Pons i Prats

Co-director: Jovana Kuljanin

Date:

Overview

Continuous growth of air traffic generates that available resources during periods of high activity cannot meet the demand of air traffic, which generates a lot of congestion. This not only affects the reduction of operational efficiency, but also increases the workload of air traffic controllers, consequently, the risk of runway incursion or collision on the taxiway increases. For this reason, it is essential to have a good method of detecting and resolving possible conflicts between vehicles.

The aim of this project is to understand all aspects that affect the generation of possible conflicts between vehicles in the airport taxi zone. To achieve this goal, a study has been carried out of the main characteristics of the airport's taxi zone: velocities, priorities, signage, among others. On the other hand, various studied and/or implemented methods of conflict detection and resolution have been analysed in order to be able to delve more deeply into the field.

In the second part of this project, a simplification of conflict detection and resolution has been created by using Python. The main objective is to be able to put in action all studied aspects of the airport in the most realistic way possible. Once done, the different trajectories and the possible conflicts generated can be visualized.

Overall, the project provides some guidelines on issues of conflict detection and resolution that are considered relevant for future research in this area. In addition, the necessary elements for implementation at the airport are mentioned.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CAGR: Compound Annual Growth Rate.

FAA: Federal Aviation Administration

ATC: Air Traffic Controller

KLM: Koninklijke Luchtvaart Maatschappij (KLM Royal Dutch Airlines)

PanAm: Pan American World Airways

ICAO: International Civil Aviation Organization

ISA: International Standard Atmosphere

AIP: Aeronautical Information Publication

A-SMGCS: Advanced Surface Movement Guidance & Control System

EHP: Extended Hybrid Petrir-nets

ADS-B: Automatic Dependent Surveillance Broadcast

ATCAM: Airport Traffic Collision Avoidance Monitor

TIS-B: Traffic Information Service – Broadcast

RSM: Runway Safety Motor

TCM: Taxi Conflict Monitor

CD&R: Conflict Detection and Resolution

VHF: Very High Frequency

ENAIRE: Spanish Airports and Air Navigation

URM: Uniform Rectilinear Motion

AI: Artificial Intelligence

INTRODUCTION

Nowadays, air transport sector is constantly growing and this trend is predicted to continue in future. IATA revealed that air transport passenger numbers could double to 8.2 billion in 2037 and over the next two decades, the forecast anticipates a 3.5% CAGR (*compound annual growth rate*), leading to a doubling in passenger numbers from today's levels. [1]
Air transport growth generates an increase of traffic of passengers causing an increase in the number of airplanes. As well as, their movements entails more vehicles on the manoeuvring area of airport in order to attend all of them. Therefore, there is an increase in potential accidents and incidents between service vehicles and aircrafts.

Actually, there is a FAA guide [2] to ground vehicles operations in order to reduce these problems but according to statistics [3] "*45% of accidents is due to of the actions of operators of ground vehicles, 77% due to of the actions of airline pilots, and 18% due to of the actions of air traffic control officers*." This should make us think that the FAA guide is insufficient in order to provide totally operational security.

The main goal of analysing the ground movements is to find routes and schedules for all vehicles moving on the airport surface in an effective manner.
Ground movement plays a key role to reduce delays because if there is a good coordination between aircrafts, service vehicles, handling and air controller, delay will be reduced. This implies that there is a good sequence between runway and the gate or stand allocation.

This project is aimed to assess how the digitalization of the airport can support and help to the reduction in the number of accidents. For this purpose, the project focusses firstly on the definition of a digital model of the airport, and secondly on the implementation of a conflict detection software. The digital model will represent the airport configuration and the vehicles movements according to their specific characteristics.
To fulfil the aims of this project it has been necessary to understand all vehicle movements in taxiways and evaluate the importance of priority as a function of type of vehicle on each taxiway.

The first part of this document provides an introduction into the airport manoeuvring area. All the types of vehicles that can be found at the airport, the priorities of the vehicles at the airport, the speed limits and the main signs necessary to learn about this project will be explained.

The second chapter is focused on understanding different conflict detection and resolution methodologies that have been studied (and/or implemented).

The third chapter introduces the conflict detection method simplification used in this project.

Finally, Chapter 4 explains the performance and results of the practical part dealing with the implementation of conflict detection and resolution on an airport.

CHAPTER 1: GENERAL INTRODUCTION TO AIRPORT MANOEUVER AREA

In the first chapter of this project, an introduction of the airport manoeuvre area will be given. Vehicles of an airport will be described giving an overview of the task of each one such as the history of the accidents on the manoeuver area of airport.

On the other hand, there will be a study of the types of conflicts and the methodologies to avoid them.

Finally priority of each vehicle in specific situations and the velocities at each area of the airport will be described.

1.1. Motivation

An airport has a complex geometry in which access must be controlled. Separation between aircraft or between aircraft and vehicles must be assure.
Most of the occurrences on airport do not have consequences in terms of loss of life but it might cause aircraft damage, delays to passengers and substantial financial costs.

One study indicates that ramp and gate conflicts are the most common accounting for almost 43%, with those on taxiways and runways representing between 25% and 30% of the reported conflicts [4].

In this section, collisions and conflicts on airport will be analysed.

1.1.1. Rodeo's accident

On 27th of March of 1977, the worse accident of aviation history occurred in which 583 people die [5]. That day, the airport was collapsed with aircrafts parked on the unique taxiway because of Gran Canaria airport was close due to a bomb warning causing many flights to be diverted to Los Rodeos.

The main reason of the accident was a communication failure between ATC (Air Traffic Controller) and the pilot. The pilot understood that he could take-off when the order was the opposite. This was due to the use of the words "authorise" and "take-off" which resulted in a misunderstanding. On the other hand, another key factor was the lack of visibility due to the weather conditions: cloudy day with fog. For this reason, only the radio could be used as a means of communication.

That occurred when one of the KLM airplanes started its take-off run while a PanAm airplane was on the runway waiting to go to the taxiway. When KLM saw it, it tried to rised up in order to exceed PanAm but it ended up ramming it.

As one of the safety improvements taken from that accident, a change of the radio communications and phraseology was implemented. Nowadays, the word “authorise” and “take-off” can only be used at the time the take-off clearance is received.

1.1.2. Taxiway collisions

This section, examines collisions and near collisions while aircraft are on the airport manoeuvring areas (including taxiways and ramp areas). The main reasons of that accidents are [6]:

- Using an incorrect taxiway.
- Failing to stop at a taxiway holding point.
- Failing to stay on the surface movement control radio frequency or ground frequency as appropriate.
- Failing to obtain a clearance before entering an area subject to control.

It could be accidents of aircraft-aircraft, aircraft-vehicle, aircraft-object, vehicle-vehicle and vehicle-object. Several accidents that happened on that area are on Appendix A.

In order to ensure efficient and safe airport ground operations, separation between vehicles/aircraft must be maintained. Many times, separation is compromised, whilst all do not result in collisions, most taxiway incidents involve vehicle operators deviating from the surface movement controller’s clearance. For that reasons, it is necessary to assure the trajectories and make the job easier to movement controllers.

1.2. Type of vehicles

Ground vehicles are key to airport operations. They are responsible for providing service to each aircraft and thus, without them, flights will not be possible.
The main vehicles that compose airside of airport are the following:

- **Aircraft:** is defined as the main vehicle of an airport. On Figure 1.1, FAA classifies aircraft as a function of landing and stall speeds.

Aircraft Category	1.3 × Stall Speed (knots)	Maximum Speed (Circling Approaches) (knots)	Typical Aircraft in This Category
A	<91	90	Small single engine
B	91–120	120	Small multiengine
C	121–140	140	Airline jet
D	141–165	165	Large jet/military jet
E[a]	>166		Special military

Figure 1.1. Aircraft Categories and Landing Speeds [7]

- **Pushback:** The typical configuration of this kind of vehicles is a low-slung configuration with large engines and wheels. They do need a very high torque to move fully loaded aircraft weighing hundreds of tons.

  The design allows the tug or towbar to get underneath the nose of the airplane to attach to the landing gear and push the aircraft back from the boarding gate or from one place of the aerodrome to another. When this action is done, airplanes could keep switching off while they are being transported.

  The speed of this vehicle is 30 km/h when it does not push an aircraft and 10 km/h on the contrary.

Figure 1.2. Pushback vehicle [8]

- **Baggage tractor:** Their function is to transport passengers' baggage that has been checked in. It has a train of compartments with baggage in them.

Figure 1.3: Baggage tractor [9]

- **Catering trucks:** Their main function is to bring packaged food to the aircraft for the passengers and crew to consume during a flight.

  Catering trucks are fitted with a temperature-controlled and high-lift platform. It can be raised to the level of the aircraft cabin and trolleys containing food and beverages are rolled into the aircraft.

Figure 1.4. Catering trucks giving service to airplane [9]

- **De-icing vehicles:** De-icing vehicles are used to spread hot water mixed with retardant to avoid the formation of ice at critical points of the airplane.

- **Snow-plowing, sweeping and blowing vehicles:** Snow and winter weather can produce damages and thus paralyze operations. For this reason, airport must include snow-plowing, snow-sweeping and snow-blowing vehicles. They are used to clear the runway and apron of snow completely in a short time (10 mins on average) in order to guarantee safety and efficiency of aircraft and airport utility vehicles.

Figure 1.5. Snowplow drives at London Gatwick airport [8]

- **"Follow me" cars:** A car which the main function is showing the way around the airport to the newly arrived aircrafts. The design of this is colourful in order to make it easier for pilots to see them.

- **Apron buses:** These are used when there is not available boarding bridge. Their objective is to move people from the terminals to the aircraft and the other way around avoiding people walking on the apron.

- **Refuelling:** Aircraft fuel servicing vehicles are a critical ground support equipment. There are two types of vehicles used in refuelling:

    - Self-propelled fuel trucks: they are similar to those that supply fuel to gas stations.

    - Hydrant dispensers: they do not have tanks on their own, so they pump fuel from ground pipelines through a system of hoses and hydrants.

Figure 1.6. Hydrant dispenser refuelling vehicle [9]

- **Ground power units:** Their function is to supply electrical power to the aircraft while it is on ground with engines switched off and to the ground support equipment. They could be self-propelled or towed.

- **Fire trucks:** In order to have a rapid response in case of emergency, there are different types of vehicles. The most common on airports is the fire truck which is equipped with a water cannon on top.

  Their main characteristics are that they are fast and very robust because they can be moved easily through different types of surfaces, including rough terrain.

  They are designed specifically to fight fires that involves amounts of kerosene. For this reason, they can carry large capacity of water and foam through powerful high-capacity pumps and cannons.

- **Stair truck:** They are used, beside apron buses, when there is not available boarding bridge.

  It consists of a truck with steps mounted on the back. These are adjustable in height to fit several airplanes.

Figure 1.7. Stair truck vehicle [9]

- **Belt loaders:** These are trucks with conveyor belts on them. They are designed to load or unload baggage on the aircraft. The ramp is positioned under the aircraft and then, is raised to the level of the aircraft cargo hold.

Figure 1.8. Belt loader vehicle [9]

- **Water trucks:** Trucks which their role is to refill the aircraft water tanks with potable water. It is transported through a pump.

- **Lavatory-service vehicles:** Aircrafts is equipped with waste tanks which, during the flight, store wastewater from lavatories. Once landing, these vehicles empty these tanks and refills it with disinfecting chemicals.

- **Maintenance:** Before each flight, there are several vehicles of maintenance of the aircraft in order to ensure correct operation of all parts of the aircraft.

- **Security, police cars:** Its main function is surveillance, verification and control of facilities, vehicles, people, mail, cargo… On the other hand, they are responsible of control of explosives and they have the capability to have an immediate response to complex situation (as could be a bomb threat), among others.

1.3. Priorities

At the airport, as on conventional roads, there are defined priorities by regulations [10] . These are designed to have an ordered traffic and thus, avoid possible accidents when there is more than one vehicle on a cross point or on the road.

Vehicles that have priority signal lights on, have preference over any vehicle except airplanes in movement.

On the other hand, on restricted area of the airport, there are established the following priorities, as we can see in Figure 1.9:

1. Airplanes in movement under their own power unless they receive instructions to yield another airplane.

2. Airplanes in movement under towing or guide-vehicles.

3. Vehicles with priority signal lights on.

4. Vehicles circulating on service roads.

5. Vehicles to be incorporated at service roads.

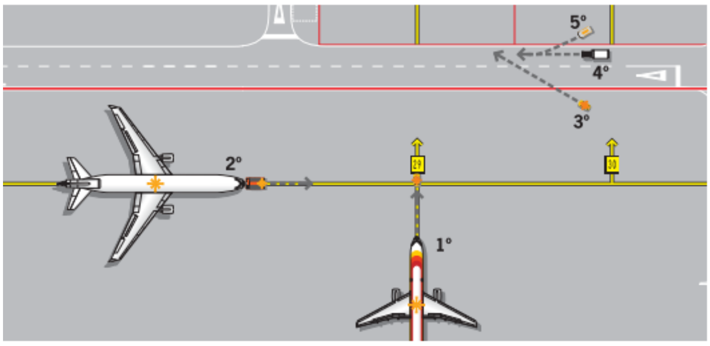

Figure 1.9. Priorities on taxiway [10]

Furthermore, as we can see on Figure 1.10, on crossing points of service roads, right takes preference unless signals or marks on road that indicate otherwise. On roundabouts, vehicles that are circulating inside have priority.

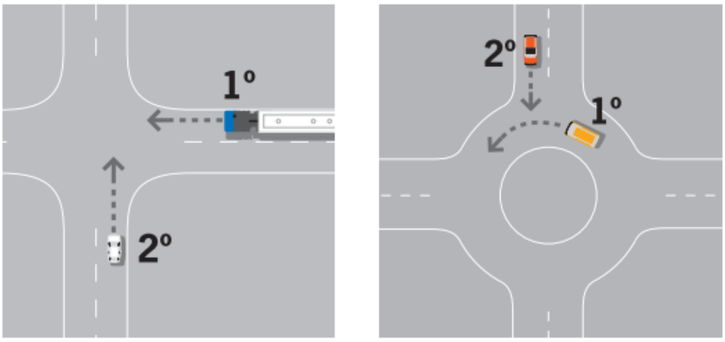

Figure 1.10. Priorities on service roads and roundabouts [10]

1.4. Velocities

Another key factor to take into account is the velocity. It is important to avoid accidents and to establish regulations that are supervised.

Velocity is defined by zones [11]:

On airport manoeuvre area, the maximum velocity is 30 km/h but there are exceptions as we can see in Figure 1.11:

- On service roads, the maximum speed is 25 km/h.

- When the vehicle enters the parking stand, the maximum velocity is 6 km/h.

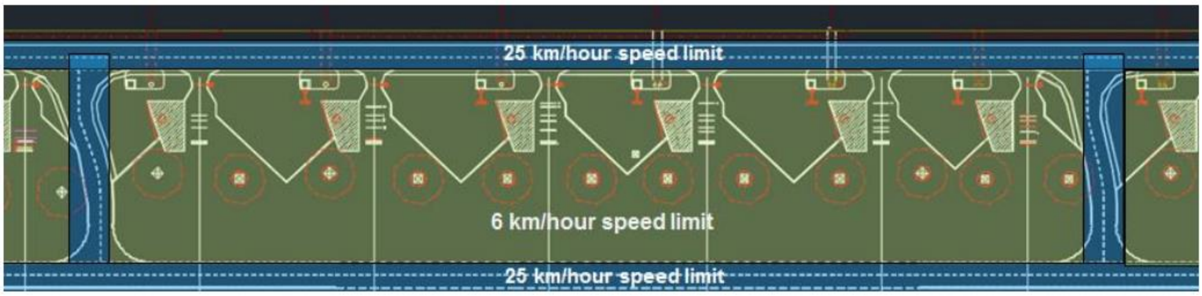

Figure 1.11. Airport parking layout showing speed limits by zones [11]

Even so, at the airport there could be signals of different maximum velocity that must be applied.

On the other hand, in exit runways, there are defined on regulation [12] the maximum velocity:

- 93 km/h (50 kt) when the reference code is 3 or 4.

- 65 km/h (35 kt) when the reference code is 1 or 2.

Key number of airport is defined on Annex 14 (ICAO) [13]. "*The intent of the reference code is to provide a simple method for interrelating the numerous specifications concerning the characteristics of aerodromes so as to provide a series of aerodrome facilities that are suitable for the aeroplanes that are intended to operate at the aerodrome. The code is composed of two elements which are related to the aeroplane performance characteristics and dimensions"*

As a function of the selected airport, we have to analyse which key number corresponds to it, the length is defined at Figure 1.12.

Code element 1	
Code number	Aeroplane reference field length
1	Less than 800 m
2	800 m up to but not including 1 200 m
3	1 200 m up to but not including 1 800 m
4	1 800 m and over
Code element 2	
Code letter	Wingspan
A	Up to but not including 15 m
B	15 m up to but not including 24 m
C	24 m up to but not including 36 m
D	36 m up to but not including 52 m
E	52 m up to but not including 65 m
F	65 m up to but not including 80 m

Figure 1.12. Aerodrome reference code [13]

The aeroplane reference field length is defined as "*the minimum field length required for take-off at maximum certificated take-off mass, at sea level, in International Standard Atmosphere (ISA) conditions in still air and with zero runway slope*". [14]

To obtain these value, through AIP, runway profiles are defined. In the case of Asturias airport, as we can see on Figure 1.13, the minimum distance for take-off with zero slope corresponds to the value 2200 m. So, the aeroplane reference field length of that airport is 2200 m. Therefore, the maximum velocity on exit runways is 93 km/h because 4 is the code number (See Figure 1.12).

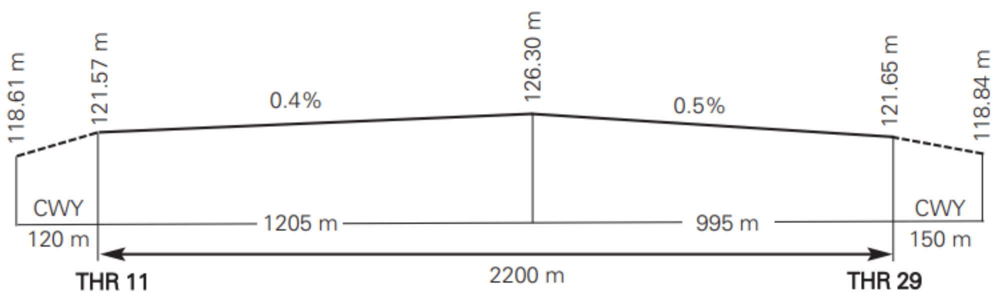

Figure 1.13. Runways profile [15]

Table 1.1 provides a summary of velocities and priorities of each vehicles as a function of a zone where it is. Aircraft assistance vehicles includes all the rest of vehicles mentioned on 1.2.

Table 1.1. Velocities and priorities of each vehicle

Type of vehicle	Zone	Velocity	Priority
Aircraft	Runway	Small single engine: 90kt	1
		Small multiengine:120kt	
		Large jet:165 kt	
	Parking stand	6 km/h	
	Exit runways	93 km/h	
	Taxiways	30 km/h	
Pushback	Parking stand	Pushing aircraft: 10 km/h	4
		Other:30 km/h	
	Other	30 km/h	
Aircraft assistance vehicles	Parking stand	6 km/h	4
	Service roads	25 km/h	
	Other	30 km/h	
Follow-me	Parking stand	6 km/h	2
	Service roads	25 km/h	
	Other	30 km/h	
Fire trucks, security, police cars	Parking stand	6 km/h	3
	Service roads	Emergency case:120 km/h Non-emergency case: 25km/h	
	Other	Emergency case:120 km/h Non-emergency case: 30 km/h	

1.5. Signals

To know where vehicle can circulate, it is necessary to observe the airport signals. There are several of these that determinates limitations as well as in terms of velocity, direction and allow to enter at zone or not.

- **Yield sign and stop strip:** The vehicle must stop and give way to the other vehicle. (See Figure 1.14)

Figure 1.14. Yield sign and stop trip [16]

- **No entry:** Does not allow a vehicle to enter an area. This sign would be located on a taxiway intended to be used in one direction only or at the intersection of vehicle roadways with runways, taxiways, or aprons where the roadway may be confused [17]. (See Figure 1.15)

Figure 1.15. No entry signal [17]

- **Closed runway:** It must not enter to the runway. There are signals to permanent closed and to temporaly closed. (See Figure 1.16)

Figure 1.16. Closed runway [18]

- **Velocity signals:** As the conventional roadways, it has limitation on speed in concrete zones of the airport.

- **Light gun signals:** There are a tool used by Air Traffic Control Towers when no radio is equipped on aircraft (or vehicle), Communications cannot be established or during communications malfunctions. [19]

Table 1.2. Light gun signals [20]

LIGHT GUN SIGNALS			
COLOR AND TYPE OF SIGNAL	MOVEMENT OF VEHICLES, EQUIPMENT, AND PERSONNEL	AIRCRAFT ON THE GROUND	AIRCRAFT IN FLIGHT
STEADY GREEN	Cleared to cross, proceed, or go	Cleared for takeoff	Cleared to land
FLASHING GREEN	Not applicable	Cleared for taxi	Return for landing (to be followed by steady green at the proper time)
STEADY RED	Stop!	Stop!	Give way to other aircraft and continue circling
FLASHING RED	Clear the taxiway/runway	Taxi clear of the runway in use	Airport unsafe, do not land
FLASHING WHITE	Return to starting point on airport	Return to starting point on airport	Not applicable
ALTERNATING RED AND GREEN	Exercise extreme caution!	Exercise extreme caution!	Exercise extreme caution!

- **Taxiway location:** This sign (See Figure 1.17) indicates the taxiway on which the vehicle is located.

Figure 1.17. Taxiway location signal [21]

- **Destination signs:** It defines the direction and destination of that runway. Figure 1.18. shows the signal of runway exit located around the runway (G2 as an example), and the others direction signals of each taxiway.

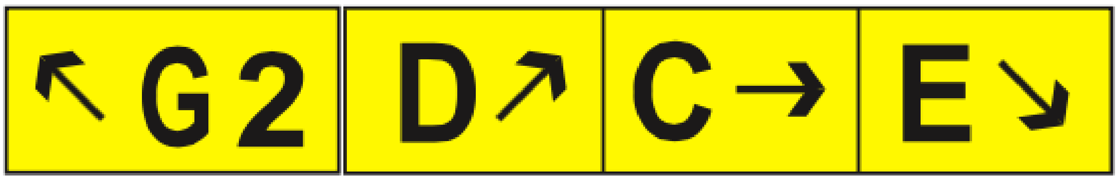

Figure 1.18. Direction signals [21]

- **Stop signal:** It is used only for auxiliary vehicles on maneuver area on the airports, in that case, airplanes are excluded [22]. The vehicle must stop and give way to other vehicles. (See Figure 1.19)

Figure 1.19. Stop signal [23]

- **Holding position markings** [18]**:**
    - Runway holding position markings: These markings indicate where the aircraft must stop when approaching a runway. These markings could be on taxiways, runways or runway approach areas. Until clearance has not been issued by ATCo, they have to be stopped. (See Figure 1.20)
    - Holding Position Markings for Instrument Landing System (ILS): When ATCo indicates to hold short of the ILS critical area, pilots must stop. (See Figure 1.20)

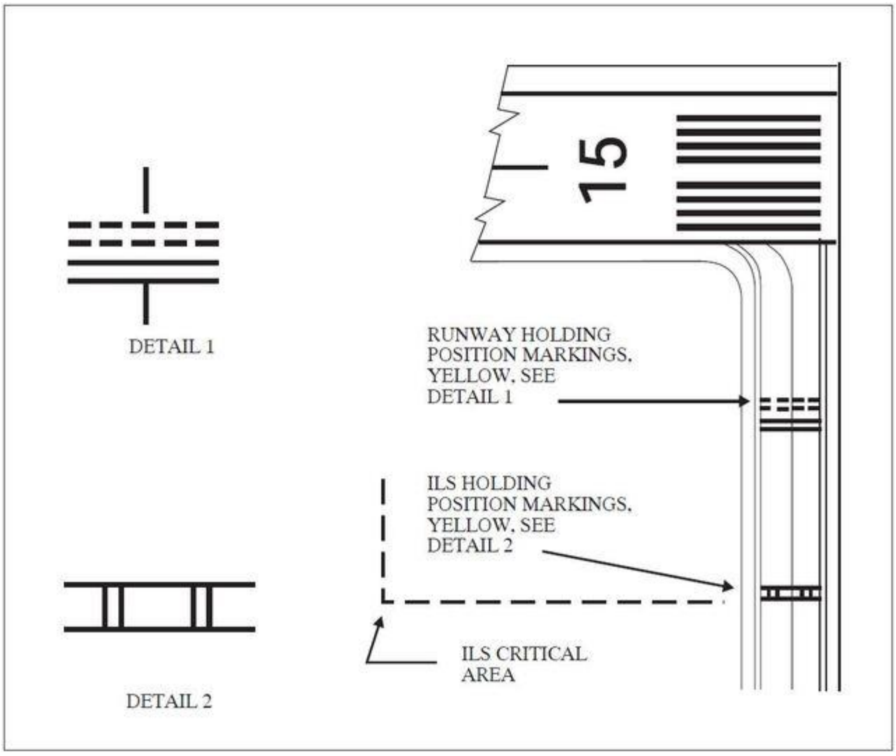

Figure 1.20. Runway holding position markings [18]

- <u>Holding Position Markings for Intersecting Taxiways:</u> When ATCo indicates: "*Hold Short of Taxiway XX"* the pilot must stop. (See Figure 1.21).

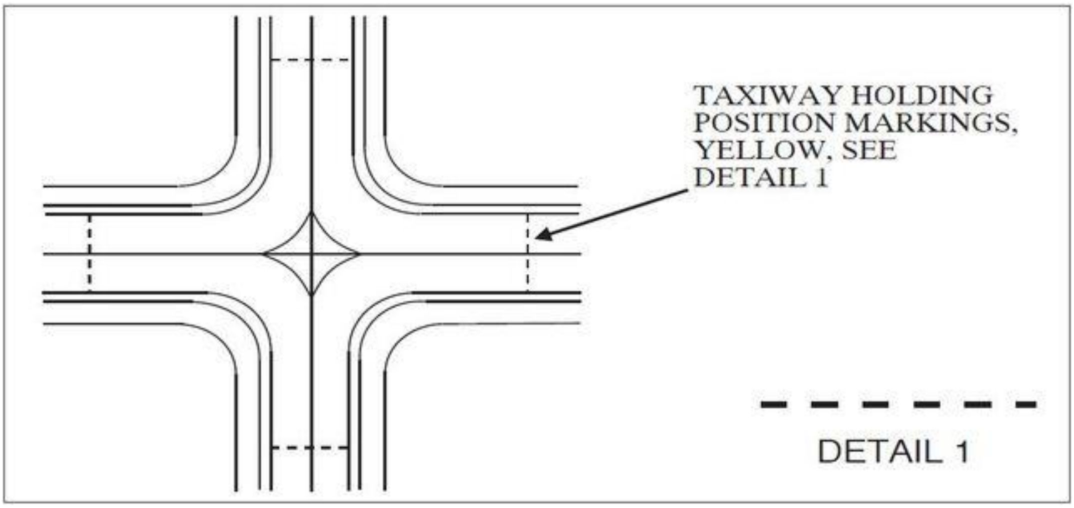

Figure 1.21. Taxiway holding position markings [18]

CHAPTER 2: CONFLICT DETECTION AND RESOLUTION METHODOLOGIES

The second chapter explains the methodologies for detecting and resolving conflicts, as well as the type of possible conflicts that can occur between vehicles. Finally, a brief conclusion will be drawn from these studies.

2.1. Types of conflict

At the airport, there could be different types of conflict as a function of the vehicle position. In this section, they will be analysed as well as the methodology that could be used in order to avoid them.

The conflicts that could appear are [24]:

- **Head to head**: It is caused when two vehicles taxi in opposite directions. To avoid this type of conflict, there are two solutions according to the priority of those vehicles: ATC can direct the less priority vehicle to stop at intersection. In case that this vehicle has passed this section, dynamic path planning will change the taxiing path of the vehicle with less priority without interrupting the taxiing of both vehicles. This type of conflict can be seen on Figure 1.11.[24]

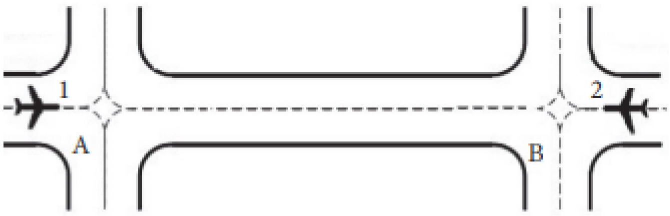

Figure 2.1. Head-to-head conflict [24]

- **Cross-conflict**: In this case, the conflict will be made on an intersection as shown on Figure 1.12. Two vehicles are expected to arrive at the same time at crossover. To avoid that, ATC can instruct aircraft of less priority to wait before reaching the intersection and after the other vehicle passes, it can continue taxing on preplanned trajectory.

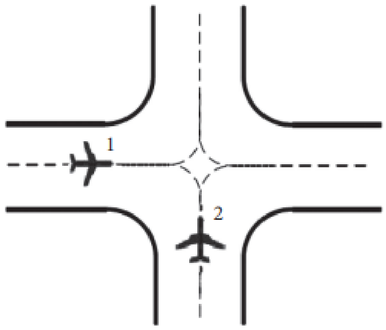

Figure 2.2. Cross-conflict [24]

- **Rear-End conflict**: Two different vehicles are riding in the same direction with different velocities. For this reason, the distance between them will be less than the minimum safety distance. This distance depends on [25]:
    - If the vehicle is in front of starting jet, it has to maintain an 8.5 m.
    - Behind the running engines of stationary aircraft, a safety distance of 50 m (conventional) or 75 m (wide body) has to be maintained.
    - Behind moving aircraft, a minimum distance of 75 m must be maintained (conventional aircraft), 125 m (wide-body aircraft with 2 engines) or 150 m (wide-body aircraft with more than 2 engines).
    - When an aircraft is moving (approaching) and it is at a distance of less than 200 m, the vehicle has to stop in order to give way to the airplane.

It may cause a rear-end conflict as shown in Figure 2.3. To avoid it, ATC may command the second vehicle, not enter in that section until the other one passes this section and meet the safety distance.

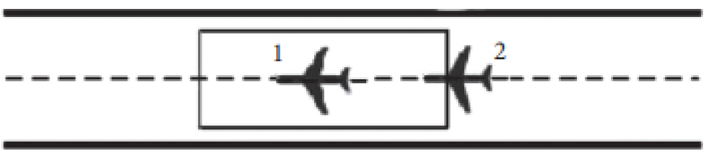

Figure 2.3. Rear-End conflict [24]

2.2. Deconflict taxiing methodologies

Actually, the conflict alerts transported are based on transponder [4]. It is an avionic system that is located on board. It provides information about aircraft as identification, barometric altitude, among others and produces a response when it receives a radio-frequency interrogation. This system is based on responses through mode A, C and S [26]:

- Mode A: Transmits an identifying code.

- Mode C: ATCo is able to see the aircraft altitude or flight level automatically.

- Mode S: Altitude capability and data exchange.

Due to the large uncertainty of that systems, in high-density environments, they have limited reliability because of high rates of nuisance and missed alerts. [4]

In this section, methodologies studied in order to execute them and the actually methodology implemented are explained.

According to ICAO [27], aircraft taxi collision conflict detection and resolution, has to comply three requirements:

- Providing conflict alert mechanism of two levels (collision pre-alerting and warning).

- Ensuring air traffic controllers and pilots have enough response time.

- Disturbance of false alarm to normal work of air traffic controllers and pilots should be avoided.

The methodology often used to avoid conflicts is waiting or changing paths by comparing priorities [24][28]:

- High-priority: It can pass directly through the conflict area according to the original path.

- Low-priority: It has to slow-down in advance and wait before entering the conflict zone or choose another sub-optimal alternative taxiing path. In this case, the objective is minimizing the time delay of this re-drive.

The main drawback of that method is that it does not consider the difference in total taxiing time that may be caused. This time corresponds to a delay for that flight (even others).

Studies of methods of conflict detection and resolution on ground have been explained in the following subsections.

2.2.1. Use of A-SMGCS and EHP model

A-SMGCS(Advanced Surface Movement Guidance & Control System) [29] consists of a surveillance system that provides the position, identification and routing of mobiles. Once the position is estimated, EHP (Extended Hybrid Petri-nets) model implementation is made.

For the detection of the possible conflicts, discrete and continuous pre-alerting are implemented through EHP. Furthermore, navigation lights are used to execute resolution strategies.

The control mechanism used to detect and solve the possible conflicts is on Figure 2.4.

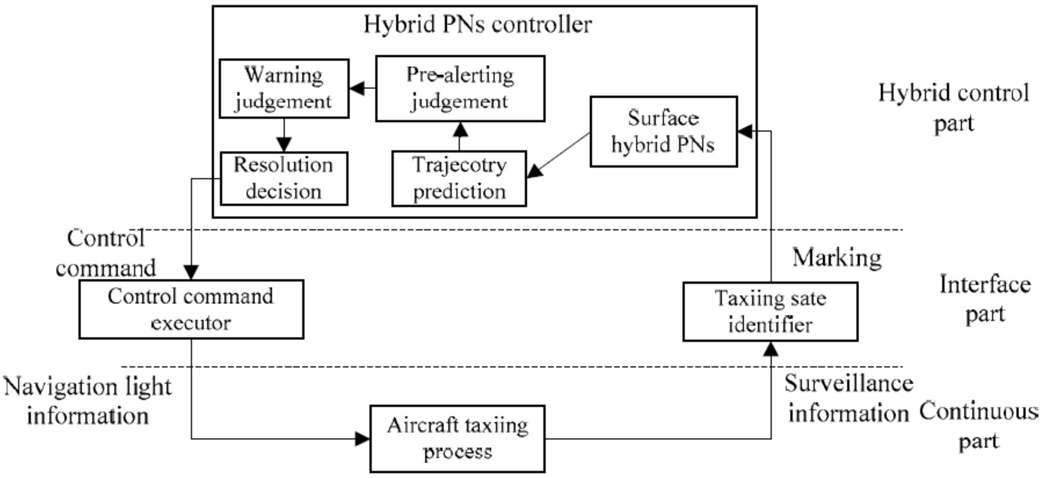

Figure 2.4. Control mechanism for A-SMGCS aircraft taxiing collision conflict detection and resolution [29]

During taxi operation, the information detected by vigilance equipment as Automatic Dependent Surveillance Broadcast (ADS-B) is processed by Petri's hybrid model.

Petri-nets model implements decision-making algorithms (including trajectory prediction algorithm, pre-alerting and warning algorithm, and resolution decision algorithm). Navigation lights of taxi-zone are switched according these control commands. In this way, conflicts could be avoided.

2.2.2. Airport Traffic Collision Avoidance Monitor (ATCAM)

It is developed for NASA by Lockheed Martin [30]. It detects potential traffic conflicts at low altitudes near the airport, on the runway and during taxi and ramp operations. [31]

There are two types of alerts:

- Caution: Requires subsequent response of flight crew but they have to be aware.

- Warning: Requires an imminent response of flight crew.

In this algorithm, the pilot reaction delay is taken into account (5 seconds when on approach, 3 seconds when rolling-out and 2 seconds during taxi).

It is composed of three separated aircraft-based algorithms which information is extracted from sources like ADS-B or TIS-B (Traffic Information Service – Broadcast). According to Eurocontrol [32], it is "*a Surveillance technique that relies on aircraft or airport vehicles broadcasting their identity, position and other information derived from on board systems"*.

These algorithms are:

- The Runway Safety Motor (RSM): It is based on runways conflicts. RSM monitors a virtual three dimensional zone around de runway, defining the positions within this zone, separation and closure rate used to determine if there will be an alert.

- The Low Altitude Conflict Monitor (LACM): It is designed to detect a possible conflict on air (at altitudes near the airport, below 1000 ft). It computes closing speed, time to closest point of approach, time to co-altitude, similar to the T-CAS approach.

- The Taxi Conflict Monitor (TCM): Designed to ground taxi conflicts. It is similar to LACM.

Then, these algorithms are integrated in order to improve the detection and the resolution methods.

Finally, to determine if there is a near collision or collision, they establish the limits based on distances that is shown on Figure 2.5**.**

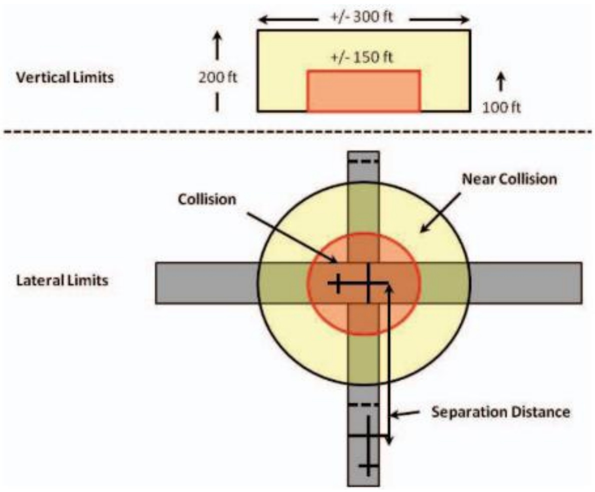

Figure 2.5. Near Collision *vs* Collision distances [30]

On the other hand, NASA have made a study [30] in order to determine the accuracy of this algorithm and the possibility of collision if an airplane is equipped with CD&R (conflict detection and resolution).

As we can see in Figure 2.6 most collisions could be avoided when both aircraft were equipped with CD&R.

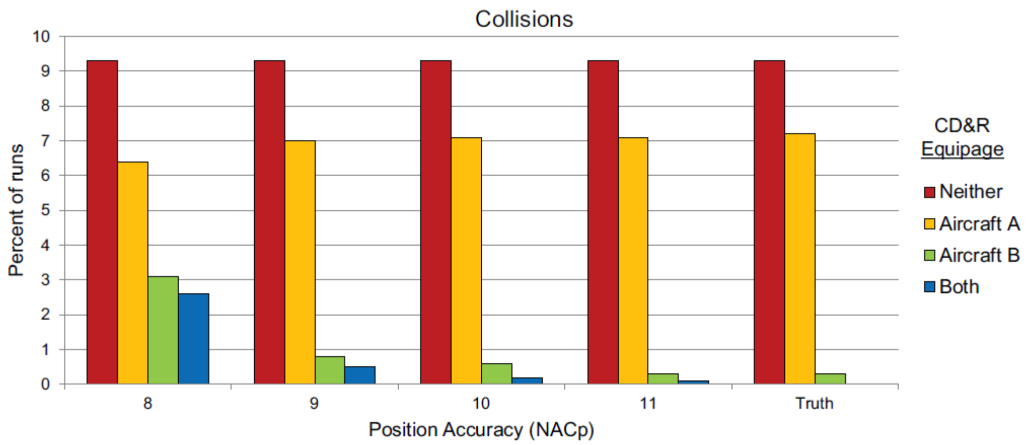

Figure 2.6. Collision Avoidance [30]

2.2.3. Conflict alerting by ADS-B

Actually, at airport most vehicles and aircraft are equipped with ADS-B. This advanced system is able to determine the position and the velocity of an airplane more accurately than radar. On the other hand, this system shares the information (as for example, the airplane type) of that vehicle to those around, including ATC. [4].

To display real-time aircraft flight tracking or vehicle tracking at the airport, there is an ADS-B based service, which could be used to implement an airport deconfliction method. It is flight radar application. [33]

As shown on Figure 2.7, through this application there is the possibility to know the path performed by each vehicle, among other characteristics: flight data by airline, aircraft, aircraft type, area, airport, velocity, etc.

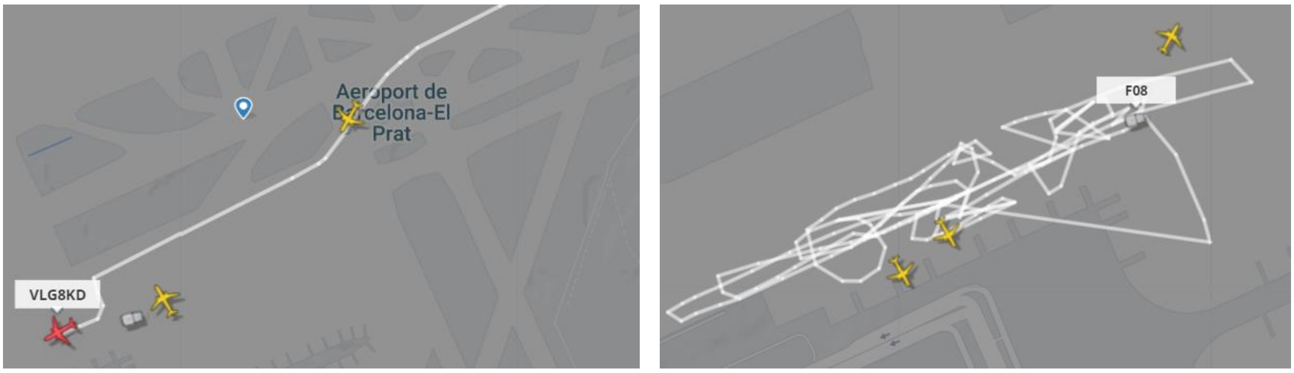

Figure 2.7. Vehicles path on Flightradar

ADS-Bs have the advantage that they update their data with much more frequency than most radars. This value corresponds to 4,8 seconds in terminal area and 12 seconds in air space en route [4].

The alert methodology proposed, evaluates different scenarios in which conflicts could occurs and, through analysing the way to transmit data of ADS-B, differents solutions are proposal [4]:

1. **Aircraft pushing back**: In this step of time, during inactivity moments, ADS-B transmits data each 5 seconds instead of twice per second (when the aircraft is moving). This change could be used to alert others vehicles that they must yield to this aircraft.

2. **Aircraft parking**: During aicraft parking, the area must be free of vehicles. If vehicle operators or flight crew could be alerted if a vehicle is predicted to be within certain dimensions. In this way, they could anticipate.
   ADS-B has a field that provides the size category of the aircraft.

   It is possible that some airports may not have the space available between the gates to allow vehicles completely parking of their aircraft footprint size. In that situation, it may be feasible to use the 24-bit-address of the Organization of International Civil Aviation (ICAO). It is used equiped through mode S on transponder in which the stationary radar encodes this address in its radio ping to specify that it wants a response from that particular transponder, and all other transponders in the air will not flood the downlink frequency with their own responses. [34]

In this case, is used to to look up the aircraft type and draw footprint more accurate than a square (See Figure 2.8).

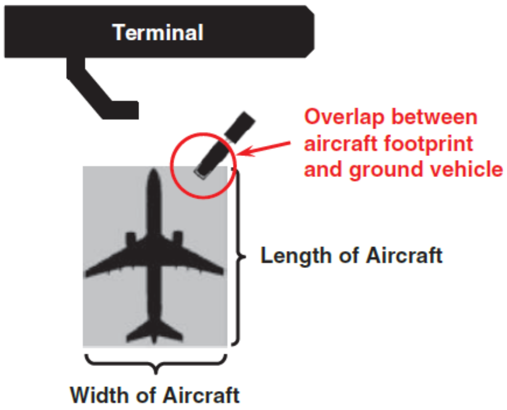

Figure 2.8. Use of aircraft length and width to identify conflicting ground vehicles during parking [4]

3. **Taxiing:** Generally, aircraft have priority over other types of vehicle during taxi operations but sometimes, it has to be possible to identify if the other ground vehicle is an emergency vehicle that actively responds to an urgency. In this case, it has right of way.
   ADS-B has a field that identifies the type of vehicle that transmits. Through this, emergency vehicle can thus be differentiated from other ground vehicles. However, to identify if it is responding to an emergency, additional information is required which may come from the priority status field in ADS-B message. This field actually is only destined to aircraft communications in case of emergency only as low fuel, radio failures,etc. The proposal is to be used also to identify ground vehicles that have priority.

4. **Runway operations:** The vehicle that is approximating to the runway must reduce velocity or even stop as it gets closer to the hold-short line. Surface vehicles are equipped with a system that alerts by the driving behavior as they approach at this line or intersection. If an aircraft is on runway, the alerting system alerts about an impending runway incursion unlessthe pattern required for a slowing vehicle can be observed in the vehicles states.

2.3. Conclusions on the actual methodology

Actually, traffic movements are mainly controlled by humans using radar displays of traffic and voice communication with pilots. The number of flights that have been controlled by one ATCo is substantially less than the number that could be encompassed by an automated air traffic control system. On the other hand, the number of mistakes would be reduced. [35]

As a summary, the main drawbacks of current research are:

- Mechanism for conflict detection and resolution have not been deeply investigated. [29]

- Conflict control does not have reliable modelling methods for surface operation. [29]

- Conflict resolution is mainly made through VHF (Very High Frequency) talk. Consequently, it generates an increase of air traffic controllers' workload. [29]

- The implementation of algorithms on the platform, needs the update of flight equipment. This involves a high cost. [29]

- Autonomous ATC System that Works to all possible traffic situations is difficult to design, implement, verify and validate to the required level of reliability and integrity. [35]

One study of NASA indicates the statistics about the consequences of the conflict between two vehicles. As seen in Figure 2.8, most conflicts take evasive action, but in second position, aircraft is damaged. This implies that it is important to have a good conflict resolution method to avoid this type of problems and decreases the possibilities of generating an important accident.

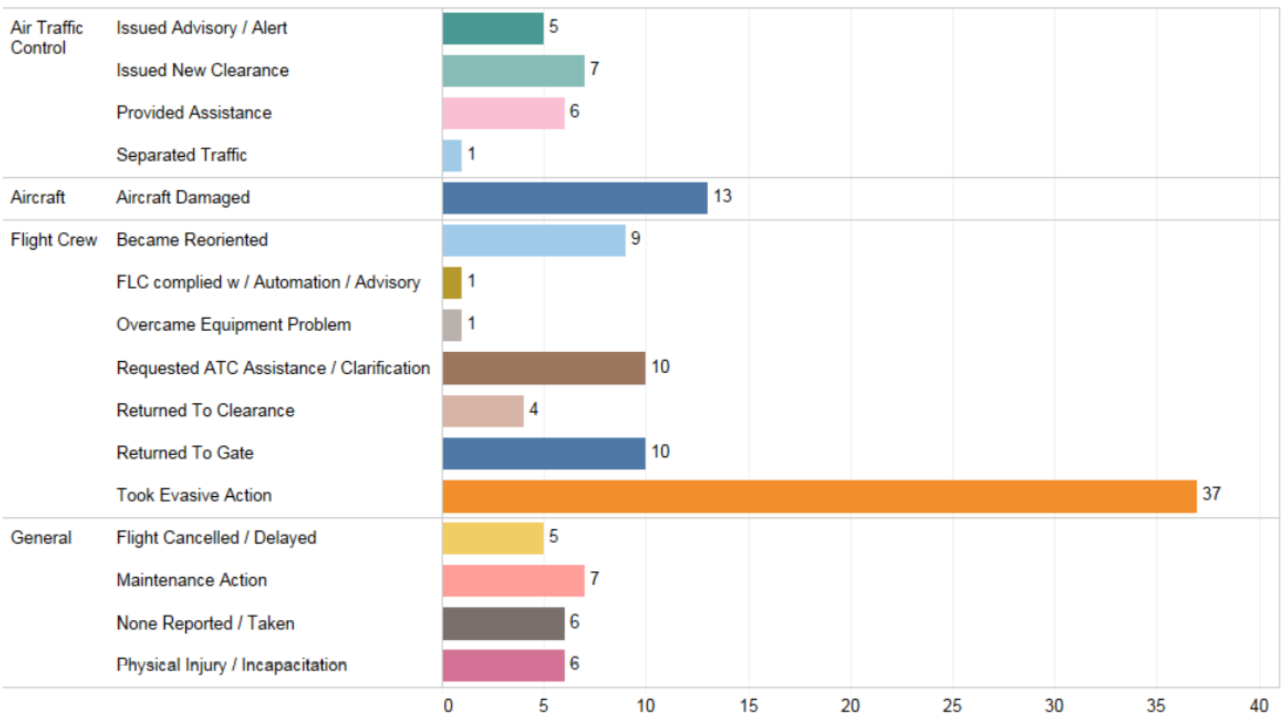

Figure 2.9. Consequences of a conflict [36]

CHAPTER 3: MODELLING OF THE PROBLEM

The third chapter is based on the modelling of the problem. The goal is to have the airside defined to make it easier to implement in code. For this, an xml file will be used in which the area of operations of the selected airport will be defined. During this chapter, the definition of this file and he phases of the digital implementation will be explained.

Furthermore, it is going to be explained how to implement the conflict detection on the airport. The main objective is to detect if the different trajectories are or not generating conflict between them.

3.1. Airport selected

In order to ease the implementation and have a good visualization of the problem, the geometry must be simple. In this way, there will be few vehicles on taxi zone.

The airport selected is Asturias. It is characterized by having a simple geometry: it has a unique runway and few taxiways. It is considered as a growing airport [37]. According to *Aena*, in 2019 it has registered 1,417,912 [38] that represents 1.3% growth with respect to the previous year.
Figure 3.1 shows the airport plan. This image shows the layout of the taxiways of the airport. It must be necessary in order to define all paths for the simulation.

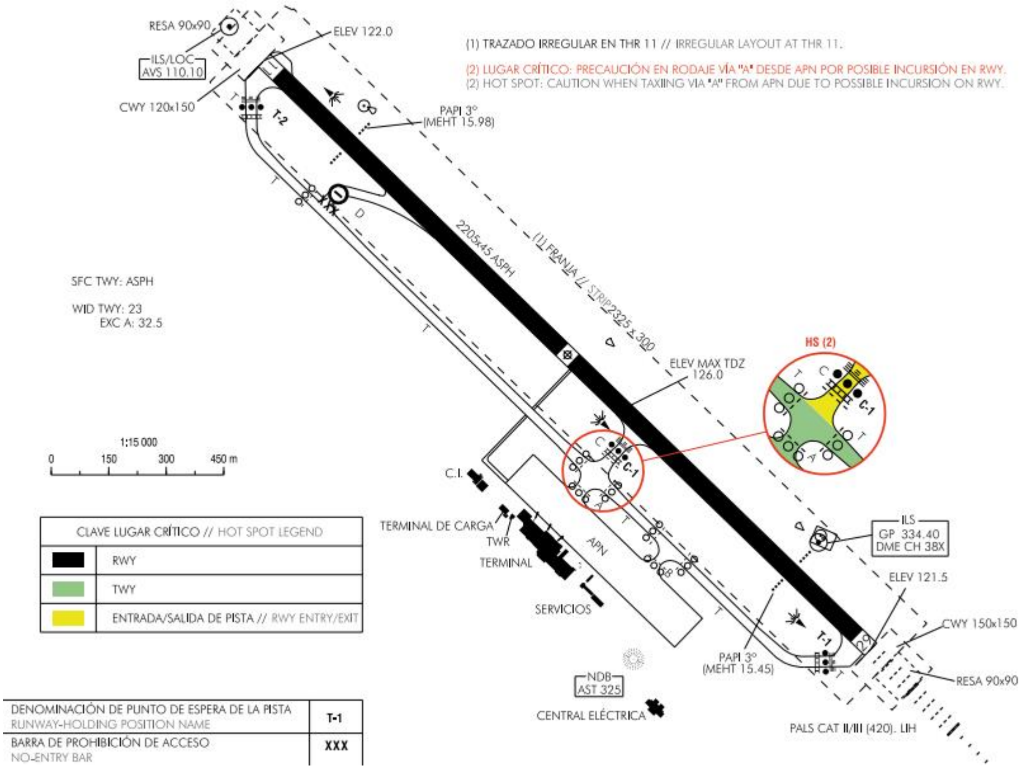

Figure 3.1. ENAIRE - Asturias Plan [15]

3.2. Airside definition

The next step is to define our airside in order to implement a simulation. The design of the airport is made through different stages as it can see in Figure 3.2.

The second stage, has been explained in Chapter 1.

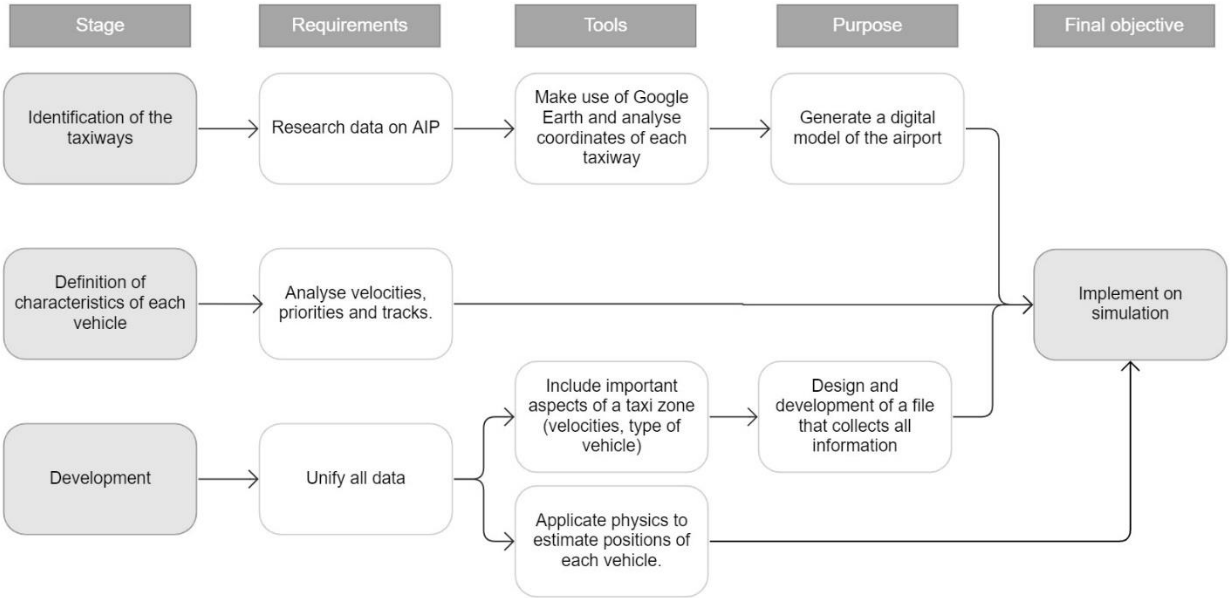

Figure 3.2. Scheme of steps for Airside Definition

3.2.1. Identification of the taxiways

The first stage is to identify each taxiway in order to characterize it (obtain its name, the type of vehicles that roads on it, the maximum speed at which it can circulate…).

Once we have identified them, it is necessary to know each coordinate where they are located. To do it, start and end points of each streetTERMINO will be established as a reference point. In addition, in intersections and curves we will locate intermediate points as we can see on Figure 3.3. The aim of that is to improve the shape definition.

Figure 3.3. Definition of curves

Finally, once all points have been defined, we obtain Figure 3.4.

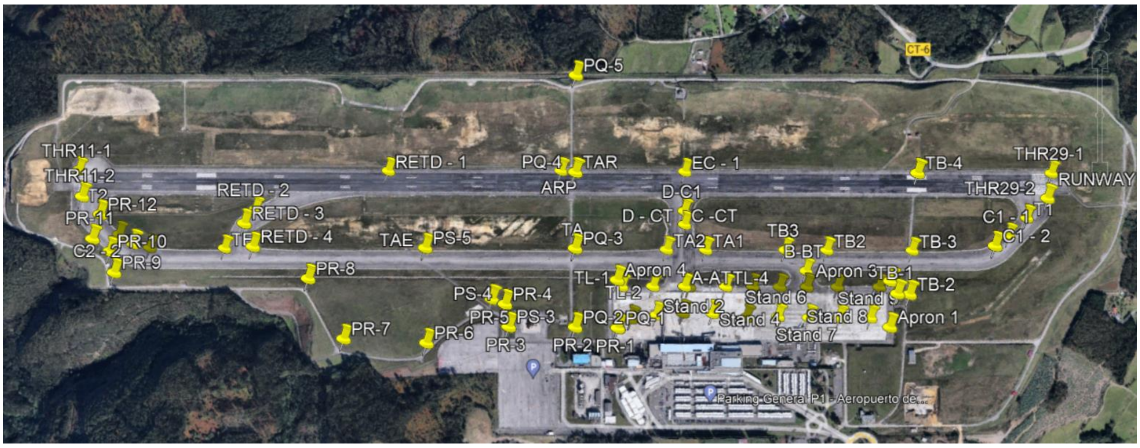

Figure 3.4. Airside definition of Asturias airport

The next step is to create the xml file. It is a document that could be read through our computer program. An extract of this file is attached at Appendix B.

3.2.2. Development

In this stage, it has to unify all data previously obtained in order to implement all of them on simulation.

First of all, the program has to read xml file and define paths of each vehicle. In that case, the trajectory is defined manually through taxiway names.

Afterwards, as a function of the type of vehicle and taxiway type, the velocity has to be defined. We assume that vehicles do not have acceleration and for this reason, they circulate at constant velocity.

In order to create different paths and be able to simulate them, it is compulsory to make a discretization. Discretization is the process through which we can transform continuous variables, models or functions into a discrete form [39]. In Figure 3.5, there is an example of a discretized representation of trajectory paths made with programming.

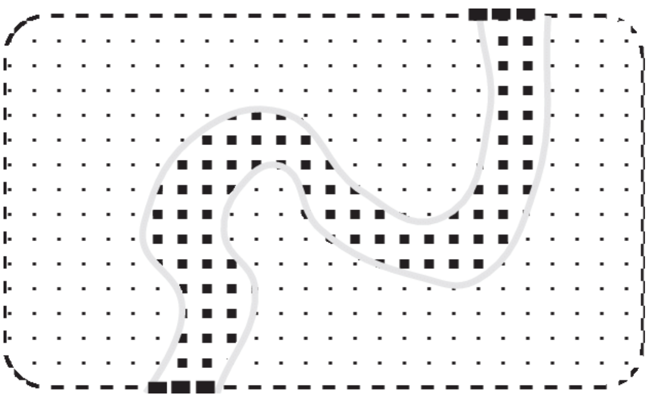

Figure 3.5. A discretized representation [40]

In that case, the discretization is made by distance. As a function of that, it is defined one, two or three points:

- If the distance ≥ 80 m, it is discretized by 4 points.

- If the distance ≥ 50 m and < 80 m, it is discretized by 2 points.

These conditions have been chosen once it has been measured the length between different points on taxiways. If we want more accuracy, we need to discretize in more points.

The next step is estimating the array of time used on programming, MRU (Movement rectilinear uniform) formulas will be implemented as we can see on **(3.1)**:

$$\boldsymbol{v} = \frac{\boldsymbol{x}}{\boldsymbol{t}} \rightarrow \boldsymbol{t} = \frac{\boldsymbol{x}}{\boldsymbol{v}} \quad \textbf{(3.1)}$$

Where x is the distance between two points that is calculated by the formula (3.4) [41] and v the velocity of the vehicle.

$$a = \sin^2\left(\frac{\Delta lat}{2}\right) + \cos(lat1) \cdot \cos(lat2) \cdot \sin^2\left(\frac{\Delta long}{2}\right) \quad \textbf{(3.2)}$$

Where Δlat = lat2 – lat1 and Δlong = long2 – long1 (of desired points).

$$\boldsymbol{c = 2 \cdot \tan2^{-1}(\sqrt{a}, \sqrt{(1-a)})} \quad \textbf{(3.3)}$$

In that case we have to select the shortest distance over the earth's surface.

$$\boldsymbol{d = R \cdot c} \quad \textbf{(3.4)}$$

R is the Earth's radius: 6372.795 km.

Figure 3.6 presents the code to calculate distance between two points.

```
distance = []
# convert to distance
i = 0
while (i + 1) < len(Fueltrajectory_x):
    rad = math.pi / 180
    dlat = Fueltrajectory_x[i + 1] - Fueltrajectory_x[i]
    dlon = Fueltrajectory_y[i + 1] - Fueltrajectory_y[i]
    R = 6372.795477598
    a = (math.sin(rad * dlat / 2)) ** 2 + math.cos(rad *
Fueltrajectory_x[i]) * math.cos(
        rad * Fueltrajectory_x[i + 1]) * (math.sin(rad * dlon / 2)) **
2
    d = 2 * R * math.asin(math.sqrt(a))
    distance.append(d)
    # time.append((d/ServiceRoadsVelocity)*3600)
    i = i + 1
```

Figure 3.6. Calculate distance between two geo points on Python

Finally, after applying URM formulas, an array of time for each vehicle is obtained.

CHAPTER 4: CONFLICT DETECTION

In this chapter, conflict detection method on airport is presented, explaining in detail different types of conflicts and the resolution of them that have been implemented.

4.1. Conflicts on Asturias airport

In this section, Asturias airport conflicts will be analysed taking into account the main types that can be found on each zone.

Initially the method to detect possible conflicts on that airport consists of the definition of different radius that will be explained in the next chapter. Once the conflict is detected, there are several options to solve that problem as seen in Figure 4.1.

On the one hand, when there is a conflict between two vehicles, both have to stop. Then, the priority of each vehicle must be evaluated. The vehicle with the lowest priority will have to continue stopped and the other one has the possibility to continue their trajectory if the conflict will be avoided. In case that it is not possible to solve it, it will be necessary to create an alternate route to one of the vehicles (normally the lowest priority).

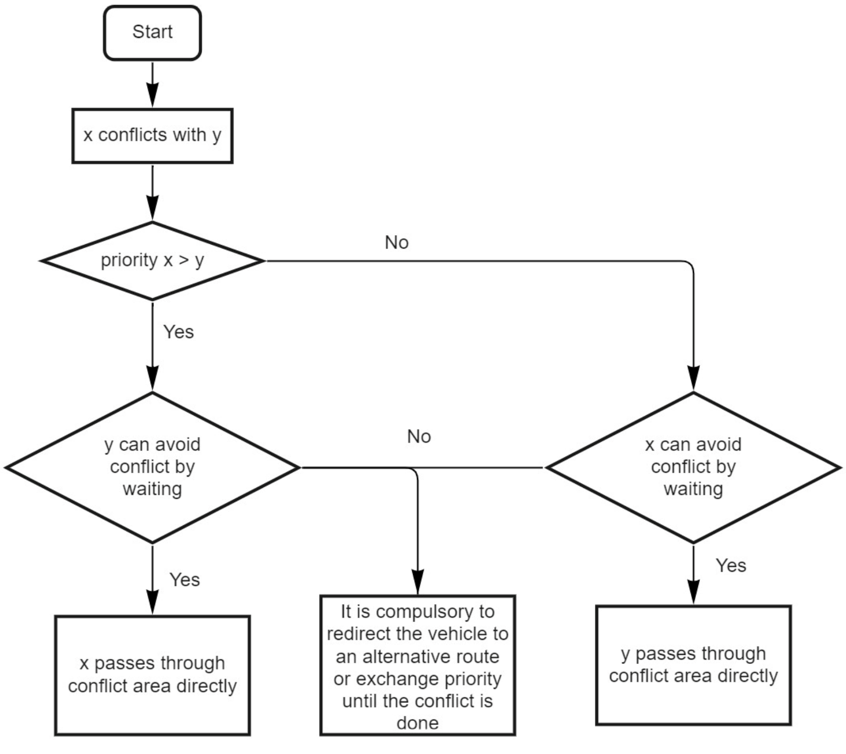

Figure 4.1. Resolution method on Asturias airport

Asturias airport has only one runway, thus generating less possibilities of head-to-head conflicts. These conflicts generally occur when aircrafts are exiting the runway.

In that case, the unique possibility of two airplanes heat-to-head could be on taxiway "T" (See Figure 3.1).

The cross conflicts are common between airplane and ground vehicles because there are several intersections between service roads and taxiways. Additionally, there are only two intersections on taxiways, thus the possibility of conflict between two aircrafts is considerably reduced.

Finally, the rear-end conflict is the most common conflict at the airport. This type of conflict could be avoided through a simply reduction of velocity.

Figure 4.2 shows the possible points of conflict at Asturias airport.

Figure 4.2. Possible points of conflict Asturias airport

On the apron zone the main conflicts are cross conflicts between airplanes. It occurs when they are entering or exiting stands. The regulation of the traffic of vehicles entering or exiting on apron zone, is the task of ATCo.

On the other hand, assistant vehicles could also generate conflicts on apron between them or between aircraft-vehicle. These conflicts could be cross, head-to head or rear-end.

In order to know the zones where each type of conflict is most probable that occurs, there are several things to taken into account:

- If the number of intersections is high, there are more possibilities that cross conflict occurs.

- Runway must be controlled, so if one vehicle desires to cross it, they have to communicate with ATCo in order to obtain permission. For this reason, the possibility that a conflict occurs is low.

- In zones with a unique lane, the possibilities of rear-end conflict increases.

- In zones with two ways, the possibilities of head-to-head conflict increases (except service roads because these ways are delimited).

On the other hand, in this section we will analyse the severity of the problem if an accident occurs in each zone:

- Runway: In this zone, airplanes, generally, go with high velocity, it implies that if there are a conflict, the possibility to avoid the problem is low and consequently, a heavy accident could happen.
  Historically, runway incursions have contributed to some of the world's worst aviation accidents. [42]

- Taxiways and service roads: The velocity is limited at airport taxi zone and service roads, so, the consequences of a possible accident are smaller than on the runway.
  On service roads, the accidents are similar to cars on the highway. Depends on type of vehicle involved in the accident, the severity of the problem could increase.

- Apron: At apron, vehicles must circulate at velocities like a person walking, which implies that the severity of the problem in case of conflict is low.

Table 4.1 contains a brief summary of the types of conflicts as a function of zones, the probability of their occurrence and the severity of the problem if it happen.

These statistics define the main type of conflict that could occur in the specific area. If there are more crossovers, the possibility of cross conflicts increases. On the other hand, on the runway and apron, head-to-head and rear-end conflict are less frequent due to the control that is carried out.

Table 4.1. Type of conflict as a function of the zone

Zone	Type of conflict	Number of intersections	Possibility	Severity of the problem
Runway	Cross	4	Low	High
	Head-to-head	-	Low	High
	Rear-end	-	Low	High
Taxiways	Cross	6	Medium	Medium
	Head-to-head	-	High	Medium
	Rear-end	-	High	Medium
Apron	Cross	12	High	Low
	Head-to-head	-	Low	Low
	Rear-end	-	Low	Low
Service roads	Cross	12	High	Medium
	Head-to-head	-	Low	Medium
	Rear-end	-	High	Medium

Note: The number of intersections in the apron zone have been calculated through the number of stands that communicate with the main road.

4.2. De-conflict method

The method of resolution of conflicts is based on the definition of a variable radius. In the next chapter, the stages followed to reach this result will be explained.

The radius will be implemented in the priority vehicle and it is a variant as a function of the near intersection. In this way, it is possible to detect cross conflicts when vehicles are distant from each other. Thereby, the conflict could be avoided with previous time and distance.

To calculate the distance from each other through code programming, formula (2.4) explained before will be used. In that case, the points are variable as a function of time. This implementation is explained in Figure 4.3.

```
def haversine(lon1, lat1, lon2, lat2):
    """
    Calculate the great circle distance between two points
    on the earth (specified in decimal degrees)
    """
    # convert decimal degrees to radians
    lon1, lat1, lon2, lat2 = map(radians, [lon1, lat1, lon2, lat2])

    # haversine formula
    dlon = lon2 - lon1
    dlat = lat2 - lat1
    a = sin(dlat/2)**2 + cos(lat1) * cos(lat2) * sin(dlon/2)**2
    c = 2 * asin(sqrt(a))
    r = 6371 # Radius of earth in kilometers. Use 3956 for miles
    return c * r
conflictpositions1x=[]
conflictpositions1y=[]
conflictpositions2x=[]
conflictpositions2y=[]

i=0
for i in range(len(trackx1)-1):
    center_point = [{'lat': tracky1[i], 'lng': trackx1[i]}]
    test_point = [{'lat': tracky2[i], 'lng': trackx2[i]}]

    lat1 = center_point[0]['lat']
    lon1 = center_point[0]['lng']
    lat2 = test_point[0]['lat']
    lon2 = test_point[0]['lng']

    radius = radiusfinal1[i]# in kilometer

    a = haversine(lon1, lat1, lon2, lat2)
```

Figure 4.3. Distance between two vehicles [43]

Once the distance is calculated (named "a" on code), it must be determined if there are conflicts between them. For it, if the condition "a" ≤ Radius, is met, the vehicle is inside the circle, so there is a conflict. In the opposite case, the vehicle is outside the area, so there is no conflict.

The code used to implement it is on Figure 4.4.

```
if a <= radius:
    print('Inside the area')
    print('Vehicle 1: ', tracky1[i],trackx1[i])
    conflictpositions1x.append(tracky1[i])
    conflictpositions1y.append(trackx1[i])
    conflictpositions2x.append(tracky2[i])
    conflictpositions2y.append(trackx2[i])
    print('Vehicle 2: ', tracky2[i],trackx2[i])


else:
    print('Outside the area')
```

Figure 4.4. Conflict detection of two vehicles

An example of conflict detection between two vehicles is shown in Figure 4.5. In this case, the orange dot corresponds to an airplane and the blue dot depicts a fuel assistant vehicle.

The radius is defined over the priority vehicle (the airplane) and the size of this radius is the distance to the near intersection. This will be the process followed to all vehicles in this simulation.

Figure 4.5. Conflict detection between two vehicles

4.3. Assumptions of the model

In these conflict detection and resolution we assume the following cases:

- The taxiing speed of all vehicles is constant. It is only dependent on the type of vehicle and the zone where it is.
- There are no accelerations. In this case, for example, on turns, there is not a decrease in velocity.
- The starting and ending points of taxiing paths of all vehicles are known.
- All vehicles have continuous taxiing paths.
- The starting time is the same for all vehicles.

In the next chapter, the limitations of these assumptions will be explained, and the possibility of implementing it on other airports.

CHAPTER 5: PERFORMANCE AND RESULTS

In this chapter, the main practical task of the present project will be exposed, an implementation of conflict detection on Asturias airport using Python. Firstly, the steps made to arrive at the desired solution including the difficulties during the implementation will be given, and then the results are going to be presented.

5.1. Digital mock-up of the airport

In this section, the simulation will be explained (the path of each vehicle and the airport definition).

5.1.1. Vehicle path definition

The first step is to define the trajectory of each vehicle. The objective is to obtain vectors with all positions of these vehicles and then, display it.

To make it, we will make use of xml created and through the *xml.etree.ElementTree* module on python, it will be possible to read it.

As explained above, the trajectories of each vehicle will be defined through calling at the function the names of each road section by section.
In Figure 5.1 there is an example of service roads extraction of the xml. It is developed in different steps (like a tree structure): firstly, locates the service roads in the file and once inside, separates them into desired section of the service roads (in this case the part defined as “entry”).

```
for serviceroads in root.iter('sr'):
    SR.append(serviceroads.attrib)
    for sr in serviceroads.iter('entry'):
        SR.append(sr.attrib)
        for latitude in sr.iter('latitude'):
            lat.append(latitude.attrib)
        for longitude in sr.iter('longitude'):
            long.append(longitude.attrib)
            for decimal_degrees in latitude.iter('decimal_degrees'):
                # print(decimal_degrees.text)
                SR.append(float(decimal_degrees.text))
            for decimal_degrees in longitude.iter('decimal_degrees'):
                #  print(decimal_degrees.text)
                SR.append(float(decimal_degrees.text))
```

Figure 5.1. Extract data from xml file

Once defined the zone, the result obtained is a list of all this section, a part of this list could be, for example:

[{'id': 'TB'}, {'id': '1'}, 43.559128, -6.026878, {'id': '2'}, 43.559039, -6.026567, {'id': '3'}, 43.559881, -6.026069, {'id': '4'}, 43.561317, -6.025244...]

To determine the zone where the vehicle could pass, it is necessary to perform a search engine in order to find the desired 'id' of the path. On Figure 5.2 it is defined the started point of that service road and through a search engine, it is possible to find them within the previously defined list.

```
initial_point = {'id': 'TB'}
subinitial = {'id': '1'}

i = 0
j = 1
encontrado = False
# initial trajectory
while (SR[i] != 0 and not encontrado):
    while SR[i] != initial_point:
        if SR[i] != initial_point:
            i = i + 1
    if SR[i] == initial_point:
        encontrado = True
        j = i + 2
while (type(SR[i]) == dict and type(SR[j]) == float):
    Fueltrajectory_x.append(SR[j])
    Fueltrajectory_y.append(SR[j + 1])
    j = j + 3
```

Figure 5.2. Path definition

5.1.2. Airport definition

Once different vehicle trajectories have been defined, the next step is to simulate them in the digitized airport. In this case, through Google Earth, airport image will be extracted.

To carry out, Matplotlib.pyplot library will be used. In this case, firstly, the program has to read the image and then the axes must be defined.
To define these axes, OpenStreetMap page [44] is used. As seen in Figure 5.3, through defining the desired area, the limits of each axis (red square) are obtained.

Figure 5.3. Axes definition [44]

Once obtain it, we extract this figure on Google Earth in order to obtain a most realistic view.

In Figure 5.4 there is the code implementation where BBox are the axes limits and ruh_m the image read of the airport.

```
for i in range(len(timegraphf)):
    BBox = (-6.0512, -6.0184,
            43.557939, 43.5689)
    ruh_m = plt.imread('airport.png')
    fig, ax = plt.subplots(figsize=(20, 8))
    ax.set_xlim(BBox[0], BBox[1])
    ax.set_ylim(BBox[2], BBox[3])
    ax.imshow(ruh_m, zorder=0, extent=BBox, aspect='auto',
interpolation='nearest')
    ax.axis('off')
    fig.tight_layout()
```

Figure 5.4. Airport simulation

5.2. Conflict detection and resolution

In this section, the conflict detection implementation will be explained as the steps to reach the desired solution.

5.2.1. Radius definition

In the first place, the idea was to implement two radius at both vehicles with their length and security range but this method, implies a lot of circles on the simulation (the possibility of false alarm increases) and the intersection between two circles is more complex to determine where the conflict is.

As we can see in Figure 5.5, there are two vehicles with their planned direction path. In this case, there is no conflict, thus having two circles, the possibilities of false alarm are increased.

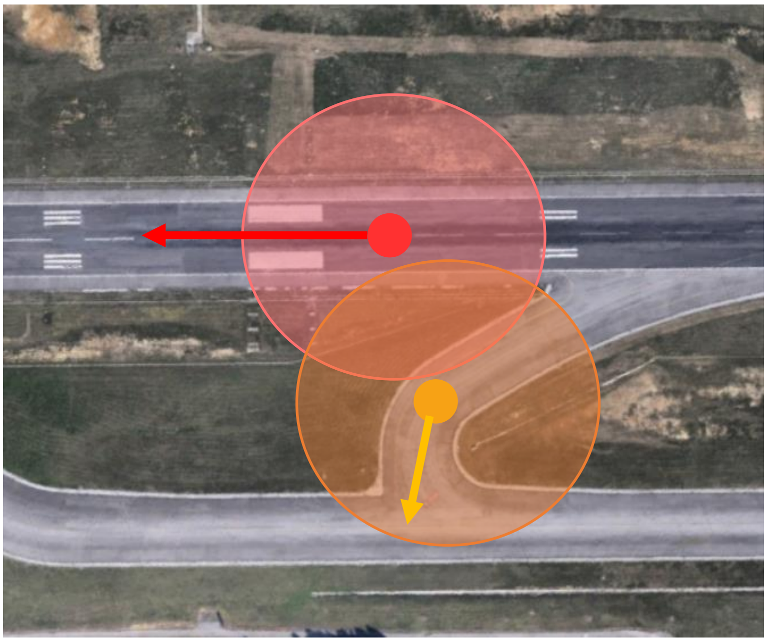

Figure 5.5. Example of circles intersection (false alarm)

In order to simplify computation, and to better detect the conflicts, the following idea was to have a fixed radius for the priority vehicle only, although the radius will depend on both vehicles.
Later it is verified that there is not manoeuvring time to avoid the problem due to the proximity of the vehicles.

This problem occurs mainly when there are intersections and it is difficult to detect a possible vehicle. In order to create a simplification to our implementation, it was decided to have a variable radius based on the nearest intersection (as have been explained in the previous chapter).

As we can see on Figure 5.6, when the conflict is detected, the vehicles are too near and the possibilities of reducing the problem are considerably reduced.

Figure 5.6. Conflict between two vehicles considering a fix radius

Finally, the chosen option has been explained at Chapter 3: The radius will be implemented in the priority vehicle and it is a variant as a function of the near intersection. In this way, it is possible to detect cross conflicts when vehicles are distant from each other. Thereby, the conflict could be avoided with previous time and distance.

5.2.2. Alerts type and resolution method

In order to detect a possible problem between vehicles well in advance, if there is more than one radius that determines the severity of the problem, the possibilities of close conflict will be reduced. For this reason, the proposal is to define two different radius:

- **Warning:** The smallest one, explained before. When there is conflict inside this circle, the vehicles must stop.

- **Caution:** The big one. As an alert, it allows us to continue the trajectory. The dimensions of that radius is twice the warning radius. Thus, the detection of the problem is with enough time.

Once the conflict is detected, the ideal is to create an alternative route if the conflict cannot be avoided, but in our case it is out of reach because the trajectories are defined manually. If we want random routes, the implementation of machine learning (AI) will be necessary.

Thus, the final solution is based on paralyzing the two vehicles and then the priority continues (if possible).

On Appendix C, there is the de-conflict method code.

5.3. **Velocities**

The velocities are defined as a function of the vehicle and the zone where it is as explained on Chapter 1.

When the vehicle path was defined, the velocity have been taken into account when time was calculated. If the vehicle changes the zone in which it is circulating (for example from a service road to a taxiway), this velocity will change.

On the other hand, the first idea was to have a deceleration on turns in order to simulate a more realistic situation but due to the implementation difficulties, the vehicles circulate at constant velocity, so, in turns, the vehicles have the same acceleration.

5.4. Difficulties with the implementation

In this section, the difficulties found during the implementation will be explained as well as the main consequences of having to do it this way.

- **Definition of routes manually**: the fact of not implementing a type of AI (artificial intelligence) or other system that helps to simplify the computation, implies a lot of time spent only in these definitions. On the other hand, it is not possible to re-route the vehicles one by one and this makes the de-conflict method limited.

- **Direction road:** The main idea was to take a look at Asturias airport signals in order to determine the direction of each road and the zones where it is possible to road or not. In AIP, there is the possibility of extracting some prohibited roads but it is not possible to see all signaling as seen on Figure 5.7. Therefore, it is necessary to visualize them through Google Earth, as an example.

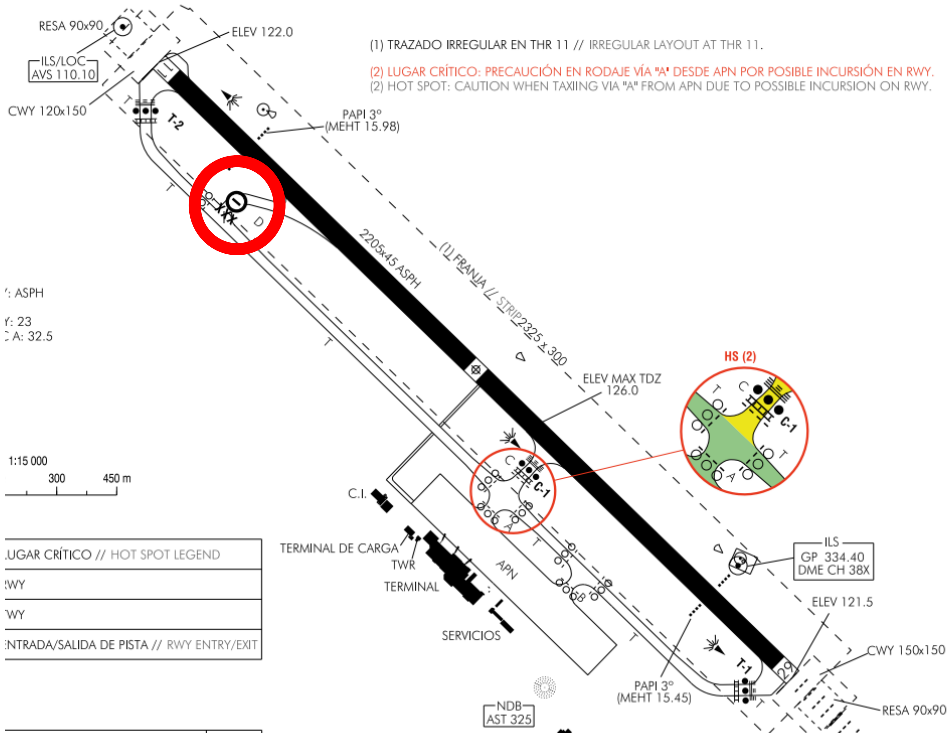

Figure 5.7. Not allowed to enter signal on Asturias AIP [15]

The problem was when looking at Google Earth, the visibility is lousy, so the definition of the direction road has been made by common sense without following any sign.

- **Resolution image of the airport:** At the moment of visualize a map on simulation, there ware several problems with the resolution of the image. Some libraries on Python, generates problems with dimensions and visualization of the points of the trajectory.

- **Making movement between points:** Firstly, there were different time vectors for each vehicle, which made it difficult to generate movement on the graph (in this case an image). For this reason, a single vector was made and movement could be generated through the plt.pause() method.

5.5. Implementation on other airports

The methodology followed could be implemented on other airports but there are many factors to taken into account:

- **The geometry of the airport**: it must be taken into account for the radius definition, if it is an airport with small airplanes, there may be nearby taxiways. This implies that a bad definition of this radius (larger) could generate false alarms, for this reason, it is important to have a good definition of it.

- **Types of vehicles**: evaluate the most common type of vehicles at the airport as well as the type of airplane it operates to simulate a more realistic situation.

- **Signals:** Try to detect the road direction and the velocity through the signals marked on the taxi zone.

- **Discretization:** Depending on the track distance, determine more or less discretization points as a function of the space.

CONCLUSIONS

With the air traffic growth, future airports will become more complex and congested. The resources available during busy periods of, generally, large airports cannot meet the demand for air transport, which causes a lot of congestion. Such congestion not only reduces the operational efficiency of airports, but also increases the workload of controllers and the risks of runway incursion or taxiway collisions. For this reason, it is important to implement an intelligent conflict detection and resolution algorithm (CD&R) which renders security surveillance more effective in detecting conflicts. [45]

The purpose of the project was to understand the methodology used for conflict detection on the airport, the type of conflict that could occur and the movement limitations that are on taxi zones. On the other hand, the creation of a simplification method in order to simulate and visualize the possible problems that can be generated. Nowadays, all objectives have been fulfilled. Along with the project, different decisions have been made in order to simplify computation and adequate this field to the academic framework, but generally difficulties have been overcome.

As explained during the project, several assumptions were considered to simplify the simulation. These could be assumed taking into account necessary factors that must be necessary in the real case: the geometry of the airport, the most common types of vehicles on that airport, signage, and velocities, among others.

Conflict detection and resolution is a field that needs in-depth research because current applications are not developed in the best desired way. In order to set CD&R algorithm on the airport zone, it is necessary to improve the vehicles equipment that implies an increase in cost.

In order to improve the current system and achieve greater security to prevent from hazardous situations and know about future traffic on the ground, it is necessary to implement a type of artificial intelligence such as Machine Learning, for example which allows to be more accurate in the prediction of results without being explicitly programmed to do so, in other words, it has an autonomous learning.

On the subject of future lines of research, there are several proposes in order to expand the project:

- Airport with more complex geometry: In this project, due to the limitation of time, the simulation have been implemented on a simply geometry

- Using ADS-B of Flight Radar: Flight Radar indicates the trajectories of vehicles that are equipped with ADS-B. A proposal to future research is to take into account these trajectories and adapt to an algorithm of conflict detection and resolution.

- Improve resolution methodology: Reduce assumptions and simplifications and try to adapt to real-case situations.

- Use of Machine Learning: It could be used in order to detect the possible conflicts and know if it is a real case or false alarm. Scikit-Learn on Python is a library for machine learning that could be applied.

Finally, the realization of this project, helps me to understand more deeply the conflict detection on taxi zone, the characteristics of that zone and of vehicles.

BIBLIOGRAPHY

[1] "IATA - IATA Forecast Predicts 8.2 billion Air Travelers in 2037." https://www.iata.org/en/pressroom/pr/2018-10-24-02/ (accessed May 29, 2022).

[2] FAA, "FAA Guide to Ground Vehicle Operations," *Fed. Aviat. Adm.*, 2015, Accessed: Apr. 22, 2022. [Online]. Available: https://www.faa.gov/airports/runway_safety/media/ground_vehicle_guide_proof_final.pdf.

[3] A. V. Shvetsov, "Analysis of Accidents Resulting from the Interaction of Air and Ground Vehicles at Airports," *Transp. Res. Procedia*, vol. 59, pp. 21–28, 2022, doi: 10.1016/J.TRPRO.2021.11.093.

[4] F. Kunzi, "Reduction of collisions between aircraft and surface vehicles," *Transp. Res. Rec.*, no. 2325, pp. 56–62, 2012, doi: 10.3141/2325-06.

[5] "45 años del accidente del aeropuerto de Los Rodeos en Tenerife - RTVE.es." https://www.rtve.es/noticias/20220328/45-anos-del-accidente-del-aeropuerto-rodeos-tenerife/2323841.shtml (accessed Jun. 08, 2022).

[6] "Taxiway Collisions | SKYbrary Aviation Safety." https://skybrary.aero/articles/taxiway-collisions (accessed Jun. 09, 2022).

[7] A. G. De Barros, "Airport planning and design," in *Routledge Handbook of Transportation*, 2015, pp. 257–274.

[8] "The ground vehicles you see at airports, explained." https://thepointsguy.com/news/the-ground-vehicles-you-see-at-airports-explained/ (accessed Apr. 28, 2022).

[9] "Airport Ground Support Vehicles - Team-BHP." https://www.team-bhp.com/forum/commercial-vehicles/226868-airport-ground-support-vehicles.html (accessed May 08, 2022).

[10] "Normativa de Seguridad en plataforma," 2017, Accessed: May 08, 2022. [Online]. Available: https://aeropuertodebarcelona.net/index_archivos/documentos/descargas/tema_aeropuertos/PCP.pdf.

[11] D. Alonso Tabares, F. Mora-Camino, and F. Mora-Camino, "Aircraft ground handling: Analysis for automation," 2017, doi: 10.2514/6.2017-3425ï.

[12] C. De Rodaje, "Oficina Regional NACC de la OACI Página 1 ORGANIZACIÓN DE AVIACIÓN CIVIL INTERNACIONAL Agencia Especializada de las Naciones Unidas," Accessed: May 09, 2022. [Online]. Available: https://www.icao.int/NACC/Documents/Meetings/2014/ANEX14SJ/D2-2.pdf.

[13] "Volume I Aerodrome Design and Operations to the Convention on International Civil Aviation Aerodromes International Standards and Recommended Practices," 2018, Accessed: May 22, 2022. [Online]. Available: https://www.iacm.gov.mz/app/uploads/2018/12/an_14_v1_Aerodromes_8ed._2018_rev.14_01.07.18.pdf.

[14] "ICAO Aerodrome Reference Code | SKYbrary Aviation Safety." https://skybrary.aero/articles/icao-aerodrome-reference-code (accessed

Jul. 25, 2022).
[15] "AIP AD 2-LEAS ADC 1.1," [Online]. Available: https://aip.enaire.es/AIP/contenido_AIP/AD/AD2/LEAS/LE_AD_2_LEAS_ADC_1_en.pdf.
[16] "Señalizaciones en plataforma de un aeropuerto." https://www.avsaf.es/post/señalizaciones-en-plataforma-de-un-aeropuerto (accessed Jul. 21, 2022).
[17] "ICAO RECOMMENDED AIRPORT SIGNS, RUNWAY AND TAXIWAY MARKINGS," [Online]. Available: http://ww1.jeppesen.com/documents/aviation/business/ifr-paper-services/airport-signs.pdf.
[18] "Airport Marking Aids and Signs." https://www.faa.gov/air_traffic/publications/atpubs/aim_html/chap2_section_3.html (accessed Jul. 21, 2022).
[19] "Light Gun Signals." https://www.cfinotebook.net/notebook/flight-hazards-and-safety/light-gun-signals (accessed Jul. 21, 2022).
[20] "Rear Cockpit: Light Gun Signaling - KITPLANES." https://www.kitplanes.com/rear-cockpit-light-gun-signaling/ (accessed Jul. 21, 2022).
[21] "Airport Signs Runway & Taxiway Signs ILS Critical Area Holding Position Sign."
[22] "airport design - Do these red stop signs apply to aircraft or only to vehicles? - Aviation Stack Exchange." https://aviation.stackexchange.com/questions/77241/do-these-red-stop-signs-apply-to-aircraft-or-only-to-vehicles (accessed Aug. 07, 2022).
[23] "Rossiya Airlines to move from Vnukovo to Sheremetyevo - Russian aviation news." http://www.rusaviainsider.com/rossiya-airlines-move-vnukovo-sheremetyevo/ (accessed Aug. 07, 2022).
[24] N. Zhao, N. Li, Y. Sun, and L. Zhang, "Research on Aircraft Surface Taxi Path Planning and Conflict Detection and Resolution," *J. Adv. Transp.*, vol. 2021, 2021, doi: 10.1155/2021/9951206.
[25] S. A. Aena S.M.E., "Normativa de seguridad en plataforma en plataforma," 2021.
[26] "Transponder | SKYbrary Aviation Safety." https://skybrary.aero/articles/transponder (accessed Jul. 27, 2022).
[27] Icao, "International Civil Aviation Organization Advanced Surface Movement Guidance and Control Systems (A-SMGCS) Manual," vol. Doc.9830, Accessed: Jul. 10, 2022. [Online]. Available: http://www.icao.int.
[28] Y. Jiang, Z. Liu, Z. Hu, and H. Zhang, "A Priority-Based Conflict Resolution Strategy for Airport Surface Traffic Considering Suboptimal Alternative Paths," *IEEE Access*, vol. 9, pp. 606–617, 2021, doi: 10.1109/ACCESS.2020.3045586.
[29] X. Zhu, X. Tang, and S. Han, "Aircraft intersection collision conflict detection and resolution under the control of A-SMGCS," *Proc. 2012 Int. Conf. Model. Identif. Control. ICMIC 2012*, no. January, pp. 120–125, 2012.
[30] D. R. Jones, R. C. Chartrand, S. R. Wilson, S. A. Commo, S. D. Otero, and G. D. Barker, "Airport traffic conflict detection and resolution algorithm evaluation," *AIAA/IEEE Digit. Avion. Syst. Conf. - Proc.*, 2012, doi: 10.1109/DASC.2012.6382370.
[31] D. R. Jones and L. J. Prinzel, "COLLISION AVOIDANCE FOR AIRPORT

TRAFFIC CONCEPT EVALUATION Sharon D . Otero and Glover D . Barker , Lockheed Martin Mission Services , Hampton , VA," 2018.
[32] "Automatic dependent surveillance – broadcast (ADS-B) | EUROCONTROL." https://www.eurocontrol.int/service/automatic-dependent-surveillance-broadcast (accessed Jul. 12, 2022).
[33] "Flightradar24: Live Flight Tracker - Real-Time Flight Tracker Map." https://www.flightradar24.com/36.97,0.24/5 (accessed Jul. 27, 2022).
[34] "transponder - What is the ICAO 24-bit code used for? - Aviation Stack Exchange." https://aviation.stackexchange.com/questions/32159/what-is-the-icao-24-bit-code-used-for (accessed Jul. 28, 2022).
[35] R. A. Paielli and H. Erzberger, "Trajectory specification for terminal airspace: conflict detection and resolution," *J. Air Transp.*, vol. 27, no. 2, pp. 51–60, 2019, doi: 10.2514/1.D0132.
[36] I. F. Ops, "Part 121 Taxi / Parked Ground Conflict Related Incidents," no. March 2018, 2020.
[37] "El Aeropuerto de Asturias cierra 2021 con 300.000 nuevos pasajeros respecto al año anterior - Noticias RTPA." https://www.rtpa.es/noticias-asturias:El-Aeropuerto-de-Asturias-cierra-2021-con-300.000-nuevos-pasajeros-respecto-al-aÃ±o-anterior_111642333548.html (accessed Jun. 24, 2022).
[38] "Inicio." https://www.aena.es/es/estadisticas/inicio.html (accessed Jun. 24, 2022).
[39] "An Introduction to Discretization Techniques for Data Scientists | by Rohan Gupta | Towards Data Science." https://towardsdatascience.com/an-introduction-to-discretization-in-data-science-55ef8c9775a2 (accessed Jul. 03, 2022).
[40] "A discretized representation ~ R of a region R, and a discrete... | Download Scientific Diagram." https://www.researchgate.net/figure/A-discretized-representation-R-of-a-region-R-and-a-discrete-d-separating-path-P-in_fig3_3298418 (accessed Jul. 03, 2022).
[41] "Calculate distance and bearing between two Latitude/Longitude points using haversine formula in JavaScript." https://www.movable-type.co.uk/scripts/latlong.html (accessed Jul. 01, 2022).
[42] "Runway Safety Guide IFALPA Runway Safety Guide."
[43] "Haversine Formula in Python (Bearing and Distance between two GPS points) - Stack Overflow." https://stackoverflow.com/questions/4913349/haversine-formula-in-python-bearing-and-distance-between-two-gps-points (accessed Jul. 14, 2022).
[44] "Exportar | OpenStreetMap." https://www.openstreetmap.org/export#map=15/43.5615/-6.0278 (accessed Jul. 20, 2022).
[45] K. Kheta *et al.*, "Vision-based Conflict Detection within Crowds based on High- Resolution Human Pose Estimation for Smart and Safe Airport," pp. 1–6.

APPENDICES

APPENDIX A. TAXIWAY COLLISIONS

The appendix *"Taxiway collisions"* shows several accidents that occurs on taxi and apron zones more deeply.

1. Aircraft/aircraft conflict:

- **A343/B752, London Heathrow airport, 1995** [6]: An Airbus 340-300 had stopped on a divergent taxiway within the runway departure hold. At that moment, the wing tip collided with tail fin of another aircraft. Only the airplane was damaged with no injuries.

- **SH33/MD83, Paris airport, 2000** [6]**:** A SH330 waiting for take-off. Later, a conditional line up clearance to runway was given while MD83 was at the runway threshold. Then, SH33 began to line up unaware that an MD83 had just been cleared to take off from the full length and a collision occurred.

2. Aircraft/vehicle conflict:

- **FA50/Snow plough vehicle, Moscow airport, 2014:** A FA50 collided with a snow plough which entered to the same runway without clearance. Control was lost and all occupants died.

- **DH8C/Fuel tanker Toronto airport, 2019:** Fuel tanker did not give way on a taxiway crossing, causing direct crew and indirect passenger injuries and substantial damage.

APPENDIX B. AIRSIDE DEFINITION - XML FILE

The appendix *“Airside definition – xml file”* cointains a part of xml code used for airport definition during the project.


```
<rwy_exit id="RETD">
   <Velocity units="km/h">93</Velocity>
   <direction>
      <type>unidireccional</type>
   </direction>
   <entry id="1">
      <latitude>
         <decimal_degrees>43.56469</decimal_degrees>
      </latitude>
      <longitude>
         <decimal_degrees>-6.03933</decimal_degrees>
      </longitude>
   </entry>
   <entry id="2">
      <Velocity units="kts">10</Velocity>
      <latitude>
         <decimal_degrees>43.56477</decimal_degrees>
      </latitude>
      <longitude>
         <decimal_degrees>-6.04320</decimal_degrees>
      </longitude>
   </entry>
   <corner id="3">
      <latitude>
         <decimal_degrees>43.564550</decimal_degrees>
      </latitude>
      <longitude>
         <decimal_degrees>-6.043636</decimal_degrees>
      </longitude>
   </corner>
   <corner id="4">
      <latitude>
         <decimal_degrees>43.564044</decimal_degrees>
      </latitude>
      <longitude>
         <decimal_degrees>-6.043594</decimal_degrees>
      </longitude>
   </corner>
</rwy_exit>
<rwy_exit id="EC">
   <Velocity units="km/h">93</Velocity>
   <direction>
      <type>unidireccional</type>
   </direction>
```

```
        <entry id="1">
          <latitude>
            <decimal_degrees>43.56282</decimal_degrees>
          </latitude>
          <longitude>
            <decimal_degrees>-6.03143</decimal_degrees>
          </longitude>
        </entry>
        <entry id="2">
          <latitude>
            <decimal_degrees>43.56206</decimal_degrees>
          </latitude>
          <longitude>
            <decimal_degrees>-6.03179</decimal_degrees>
          </longitude>
        </entry>
      </rwy_exit>
    </runways>
    <taxiways>
      <Vehicle>
        <Type>Aircraft</Type>
        <Type>FollowMe</Type>
        <Type>BaggageTractor</Type>
        <Type>Fuel</Type>
      </Vehicle>

      <Velocity units="km/h">55</Velocity>
      <TWY id="T">
        <direction>
          <type>bidireccional</type>
        </direction>
        <holding_point id="T1">
          <latitude>
            <decimal_degrees>43.55979</decimal_degrees>
          </latitude>
          <longitude>
            <decimal_degrees>-6.02271</decimal_degrees>
          </longitude>
        </holding_point>
        <holding_point id="TB1">
          <latitude>
            <decimal_degrees>43.55988</decimal_degrees>
          </latitude>
          <longitude>
            <decimal_degrees>-6.02607</decimal_degrees>
          </longitude>
        </holding_point>
        <holding_point id="TB2">
          <latitude>
            <decimal_degrees>43.56042</decimal_degrees>
```

```
    </latitude>
    <longitude>
      <decimal_degrees>-6.02834</decimal_degrees>
    </longitude>
  </holding_point>
  <holding_point id="TB3">
    <latitude>
      <decimal_degrees>43.56068</decimal_degrees>
    </latitude>
    <longitude>
      <decimal_degrees>-6.02945</decimal_degrees>
    </longitude>
```

APPENDIX C. STOP VEHICLE CODE

The appendix *“Stop vehicle code”* cointains the code used for stop the non-priority vehicle during the project.

```
def stopvehicle (x1,y1,x2,y2):
    i=0
    j=0
    m=0
    x=0
    n=0
    u=0
    xfinal1=[]
    yfinal1=[]
    xfinal2=[]
    yfinal2=[]

    while i< len(x1) or m<len(y1):
        if i< len(x1) and m<len(y1):
            xfinal1.append(x1[i])
            yfinal1.append(y1[i])
            xfinal2.append(x2[m])
            yfinal2.append(y2[m])
            if (x1[i]==conflictpositions1x[j] and
y1[i]==conflictpositions1y[j]) and (x2[m]==conflictpositions2x[j] and
y2[m]==conflictpositions2y[j]):
                if priority2 =='High':
                      x = conflictpositions1x[n]
                      y = conflictpositions1y[n]
                      m=m+1
                      xfinal1.append(x)
                      yfinal1.append(y)
                      xfinal2.append(x2[m])
                      yfinal2.append(y2[m])
                      center_point = [{'lat': xfinal2[-1], 'lng':
yfinal2[-1]}]
                      test_point = [{'lat':x, 'lng': y}]

                      lat1 = center_point[0]['lat']
                      lon1 = center_point[0]['lng']
                      lat2 = test_point[0]['lat']
                      lon2 = test_point[0]['lng']
                      radius = Radius1  # in kilometer

                      a = haversine(lon1, lat1, lon2, lat2)
                      if a <= radius:
                          while a<=radius:
                            m = m + 1
                            xfinal1.append(x)
                            yfinal1.append(y)
                            xfinal2.append(x2[m])
                            yfinal2.append(y2[m])
                            center_point = [{'lat': xfinal2[-1],
'lng': yfinal2[-1]}]
                            test_point = [{'lat': x, 'lng': y}]

                            lat1 = center_point[0]['lat']
                            lon1 = center_point[0]['lng']
                            lat2 = test_point[0]['lat']
                            lon2 = test_point[0]['lng']
```

```
                            radius = Radius1   # in kilometer

                            a = haversine(lon1, lat1, lon2, lat2)

                      else:
                          i+1
                          xfinal1.append(x1[i])
                          yfinal1.append(y1[i])
                          xfinal2.append(x2[m])
                          yfinal2.append(y2[m])

            else:
                  i=i+1
                  m=m+1
                  n=j
        if i>=len(x1) and m<len(x2):
            xfinal2.append(x2[m])
            xfinal1.append(x1[-1])
            yfinal2.append(y2[m])
            yfinal1.append(y1[-1])
            m=m+1
        if m>=len(x2) and i<len(x1):
            xfinal1.append(x1[i])
            xfinal2.append(x2[-1])
            yfinal1.append(y1[i])
            yfinal2.append(y2[-1])
            i=i+1
    return xfinal1,xfinal2,yfinal1,yfinal2
```

APPENDIX D – TFG COMPLETE CODE

Find attached the complete TFG code on the repository:

https://github.com/samanthamercader/TFGcode.git