MA.004-373-1

Saad Dahleb University – Blida1 -

Faculty of Sciences
Computer Science department

جامعة البليدة - 1 -
مكتبة
الكلية
كلية العلوم

Report submitted by
Feriel Taferhit

Theme: Unmanned Aerial Vehicles and Internet of Things Simulator

In order to obtain the Master degree in Computer Science

Domain: Mathematics and Computer Science

Spinneret: Computer Science

Specialty: Software Engineering

Supervisor: Prof. Djamel Djenouri — *Senior Research Scientist CERIST*
Instructor: Fatima Zohra Djiroun — *Phd Student Blida1*
President of the jury: Prof. Mohamed Ould Khaoua

Host Organization: Aalto University School of Electrical Engineering, Finland
&
Research Center for Scientific and Technical Information CERIST, Algeria

Supported on: September 2017

MA-004-373-1

جامعة البليدة -1-
مكتبة
الكلية
كلية العلوم

Acknowledgments

I extend to my sincere thanks to ALLAH who gave me the courage and the strength to do this work and for the help throughout my life.

I would like to express my gratitude to all the people who have supported and encouraged me in this work..

First of all, I especially want to thank my supervisor Professor Djamel Djenouri for giving me the opportunity to work under his guidance, for his enthusiastic engagement in my work and his never ending support, motivation, and orientation throughout the whole research process. I retain his availability, his unfailing help, his wise advice and rich ideas, as well as his sympathy. I would like to thank him for bringing me his immense knowledge and experience throughout the period of my internship.

I thank my instructor, Ms. Fatima Zahra Djiroun,for her help in reading the report. I am grateful to Professor Menouar Boughedaoui for his contribution on the Erasmus projects, his valuable help, his encouragement and for the time he has devoted to me.

I would also like to thank my friends at CERIST, Imane and Roufaida for the presence whenever needed. Thank you for your listening, offering me advice and for your help in gathering the information for this report.

My thanks go to Professor Tarik Taleb, the director of the Mosaic Laboratory in Finland, for having accepted me in his organization and his advice during this project; and to Dr. Miloud Bagaa, for his encouragement and to allow me to benefit from a rich training. I thank Ms. Anita Bisi, Head of International Relations at Aalto University, School of Electrical Engineering in Finland for her support and advice.

I would not forget to remember Haithem Boukartaba, Noura Benoui, Bilel Benmoussa, engineers at Industrial Technologies Research Center in Bousmail for their help, availabilities and their valuables information.
My thanks also go to all those who have kindly accepted to participate in the jury of this report.

Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my life. This accomplishment would not have been possible without them.

Blida, September 07, 2017
Feriel Taferhit

Dedication

I dedicate this modest work:

In memory of my father Djamel, who is always in my mind and heart,

To the one I love, my mother Zahia, for her patience, love, support and encouragement,

To my sister Fella and her husband Nadjib who have always been by my side,

To my second mother Nassima, my aunts Zohra and Fatma,

To the people who always helped and encouraged me, who was always by my side and who accompanied me during my study path, my family and my kind friends.

Feriel Taferhit

ملخص

إنترنت الأشياء هي شبكة من الأشياء المادية المتضمنة مع الإلكترونيات، البرمجيات، أجهزة الاستشعار، وشبكة الاتصال، والتي تمكن هذه الأشياء من جمع وتبادل البيانات. إنترنت الأشياء تستخدم واجهات ذكية لربط العالم الافتراضي لتكنولوجيا المعلومات مع العالم الحقيقي للأشياء والأجهزة. لا يتم تثبيت هذه الأجهزة فقط على الأرض، بل العديد منها يمكن تركيبها في مركبات جوية بدون طيار أو الطائرات بدون طيار.

الطائرات بدون طيار هي طائرة بدون طيار بشري على متنها يتم التحكم بها من قبل الإنسان من الأرض عن بعد أو بشكل مستقل باستخدام أجهزة الكمبيوتر حيث انها توفر خدمات كثيرة من السماء حيثما كان ذلك ضروريا. فمن غير العملي القيام باختبار حقيقي قبل اكتمال التطوير الكافي او تحقيق الإصدارات الناضجة من البرمجيات و المعدات اللازمة لتصميم محاكاة لتقليد البيئة الحقيقية.

ان مشروع نهاية الدراسة الذي يندرج تحت عنوان " محاكاة المركابات الجوية المستقلة و أنترنيت الاشياء " الذي يندرج في إطار العمل البحثي في مخابر البحث على مستوى مخبرفسيفساء، جامعة فلندا و مركز البحث في الإعلام العلمي و التقني، الجزائر، الذي يهدف إلى استغلال الطائرات بدون طيار للاستفادة منها والتزود بالعديد من الخدمات ذات القيمة المضافة خاصة في مجال أنترنيت الاشياء وأجهزة الاستشعار.

وقد ركزت مهمتنا على تصميم وتنفيذ جهاز محاكاة لاختبار وتقييم النتائج المختلفة لبعثة طائرة بدون طيار. أولا، تم دراسة مجموعة من المفاهيم المتعلقة بالمحاكاة، ثم تم تطوير، تنفيذ و اختبار وحدات مختلفة من برنامج المحاكاة المطلوب بما في ذلك تنقل الطائرات بدون طيار، جمع البيانات، والاتصالات.

وتبين النتائج التي تحصلنا عليها أن تطبيقنا يقدم أداة تساعد المطورين على اختبار جدوى الفعالية واتخاذ القرارات المستقبلية بشأن السيناريوهات المختلفة للمحاكاة قبل المضي قدما في التنفيذ.

الكلمات الرئيسية

إنترنت الأشياء، المركبات الجوية بدون طيار، طائرات بدون طيار، مهمة، التنقل، جمع البيانات، الاتصالات.

Résumé

Internet des Objets (IdO) consiste à interconnecter des objets physiques embarqués à l'électronique, au logiciel, aux capteurs . Ceci leur permet de collecter et d'échanger des données à travers internet. IdO utilise des interfaces intelligentes pour connecter le monde virtuel des technologies de l'information au monde réel des objets et des dispositifs. Ces dispositifs non seulement sont installés sur la terre, même beaucoup d'entre eux peuvent être installés en tant que Véhicules Aériens Autonomes (VAA) ou drones.

Un Véhicule Aérien Autonome est un aéronef sans pilote humain à bord. Il est contrôlé par un humain au sol par l'intermédiaire d'un dispositif à distant. Il peut également être contrôlé de manière autonome en utilisant des ordinateurs. Il fournit des services de l'Internet des Objets depuis le ciel où cela s'avère nécessaire. Il n'est pas pratique de faire un test réel avant que suffisamment de développement soit terminé et que des versions matures du logiciel / matériel des UAV soient réalisées, ce qui rend la nécessité de concevoir un simulateur pour imiter l'environnement réel.

Notre projet de fin d'étude, intitulé "Simulateur pour les Véhicules Aériennes Autonome et l'Internet des Objets" s'inscrit dans le cadre d'un travail de recherche au sein du laboratoire de recherche MOSAIC de l'université d'AALTO Finlande, et le centre de recherche CERIST, qui vise à exploiter les avantages des drones et d'en tirer bénéfice en les utilisant pour offrir de nombreux services à valeur ajoutée en particulier dans le domaine de l'Internet des Objets (IdO) et des capteurs.

Le but de notre projet est la conception et l'implémentation d'un simulateur pour tester et évaluer les différents résultats d'une mission d'un drone. Pour ce faire, nous avons étudié un ensemble de concepts liés au simulateur. Par la suite, nous avons développé, implémenté et testé les différents modules du simulateur souhaité, y compris la mobilité des drones, la collecte de données et la communication.

Les résultats montrent que notre implémentation offre un outil qui aide les développeurs à tester la faisabilité des scénarios et à prendre des décisions futures sur les différents scénarios du simulateur avant de lancer des missions ou même de faire un test réel.

Mots clé : Internet des Objets (IdO), Véhicules Aériens Autonomes (VAA), drones, mission, mobilité, collecte de données, communication.

Abstract

Internet of Things (IoT) is the interconnection of physical objects embedded with electronics, software, sensors, network connectivity which enables these objects to collect and exchange data over internet. IoT utilizes the intelligent interface to connect the virtual world of information technology with the real world of objects and devices. These devices are not only installed on the earth, but many of them can be installed as Unmanned Aerial Vehicles (UAVs) or drones.

A UAV is an aircraft without a human pilot on board. It is controlled by a human on the ground via a remote device. It can also be controlled autonomously by using computers. It provides IoT services from the sky wherever needed. It is impractical to make real test before enough development is completed and mature versions of the software/hardware of the UAVs are achieve, which yields the necessity to design a simulator to mimic the real environment.

Our Master's project, entitled "Unmanned Aerial Vehicles and Internet of Things Simulator" is part of a research project within the MOSAIC research laboratory of the University of AALTO Finland, and CERIST research center, which aims to exploit the advantages of drones and to benefit from them by using them to offer many value-added services, in particular in the field of Internet of Things (IoT) and sensors.

Our project has been focusing on designing and implementing a simulator to test and evaluate the different results of a mission of a drone. First, a set of concept related to the simulator has been studied, then different modules of the desired simulator including the mobility of drones, data collection, and communication have been developed, implemented and tested.

The results show that our implementation offers a tool that helps developers to test the feasibility of scenarios and make future decisions on the different scenarios of the simulator before proceeding on to launching missions or even making some real test.

Key words : Internet of Things (IoT), Unmanned Aerial Vehicles (UAVs), drones, mission, mobility, data collection, communication.

Contents

List of Figures

List of Tables

Liste des symboles

AI	Artificial Intelligence
ALG	Application-Layer-Gateway
API	Applications Programmer Interfaces
ARQ	Automatic Repeat Request
BLE	Bluetooth Low-Energy
CSP	Constraint Satisfaction Problem
EMS	Exception Management System
FANET	Flying Ad-hoc NETwork
GPRS	General Packet Radio Service
GSM	Global System for Mobile
HMI	Human Machine Interface
HTTP	Hyper Text Transfer Protocol
IDE	Integrated Development Environment
IoT	Internet of Things
IP	Internet Protocol
LP	Linear programming
LTE	Long Term Evolution
M2M	Machine-to-Machine
MCU	Micro Controller Unit
MILP	Mixed Integer Linear Programming
MIQP	Mixed Integer Quadratic Programming
MISOCP	Mixed Integer Second Order Cone Programming
MPP	Mission Planning Problem
MTC	Machine Type Communication

MTOW	Maximum Take Of Weight
MVT	Model View Template
NFC	Near-Field Communication
OO	Object-Oriented
POS	Personal Operating Space
QCQP	Quadratically constrained quadratic programming
QoS	Quality of Service
QP	Quadratic programming
RESTAPI	Representational State Transfer Application Programming Interface
RFI	Radio Frequency Identification
RSSI	Receive-and-Send Signal Strength
SNR	Signal-to-Noise Ratio
SOAP	Simple Object Access Protocol
SOCP	Second Order Cone Programming
SQL	Structured Query Language
TSP	Travelling Salesman Problem
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
UMTS	Uniform Resource Locator
URL	Universal Mobile Telecommunication System
UWB	Ultra-WideBand
VRP	Vehicle Routing Problem
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Networks
WWAN	Wireless Wide Area Network
XML	eXtensible Markup Language

General Introduction

In this section we give a presentation of the project, the problem addressed related to this work, and the targeted objectives. We also present the environment in which our project took place, and we end by giving a sketche of the organization of this report.

1 Presentation of the Project

1.1 Description of Project

In the near future, Unmanned Aerial Vehicles (UAVs), also known as drones, are expected to be rapidly deployed in diverse sectors of our daily life, and to performing wide-ranging activities [1]. UAVs' applications can be broadly divided into civilian and military models.
Civilian applications include those dealing with for governmental or non-governmental purposes; e.g. employing UAVs in rescue operations to recover from large-scale disaster events, such as earthquakes, forest fires, tsunamis, etc. For example during the Japan East great earthquake [2], UAVs were used to :

i) Coordinate disaster relief efforts.

ii) Capture images of the damaged reactors at the Fukushima Daiichi nuclear power plant for site assessment.

iii) Provide real-time data of radiation level at the nuclear power plant.

iv) Assess the state of the cleanup and reconstruction efforts that were taking place in Fukushima prefecture [3].

In another rescue use case, UAVs are daily used to rescue migrants in the Mediterranean sea [4] with the help of video surveillance systems. UAVs were then used to distribute nutrition and medical items among the victims as well as to coordinate the operations of relief teams. Military UAVs have also been used in the past decade. Recently, a USA presidential candidate proposed using drones for monitoring USA borders as a countermeasure against illegal immigration. Drones are also used in developing countries, e.g., in the natural disasters of Indonesia [5], and the earthquake of Nepal [6]. They are expected to be largely used in the future in other developing countries such as Algeria.

Internet of Things technologies is also gaining ground with their potential on our daily life. It is expected to provision diverse civilian, commercial and governmental services.

IoT devices are small entities that are able to operate independently to ensure a specified services.

Each IoT device has a set of sensors to sense the environment, a set of actuators to manage some tasks, and a set of wireless communication access technologies to communicate with remote servers. Whilst these UAVs would be deployed for specific objectives (e.g., service delivery), they can be, at the same time, used to offer new Internet of Things (IoT) value-added services when they are equipped with suitable and remotely controllable machine-type-communications (MTC) devices (i.e., sensors, cameras, and actuators).

The aim of this project is to design a simulator that would be used later for designing productive solutions and testing them in an efficient way that reflects the real environment. This simulator should mimic the mobility patterns of different UAVs. A set of UAVs can be on the ground or flying at different speeds, altitude, and through different routes. The suggested simulator is expected to allow the addition and removal of sensors and IoT devices to UAVs as objects. Each sensor or IoT device should be characterized by a set of properties, such as the sensing rate, the energy consumption, its capability, etc.

An UAV may have on-board one or multiple sensors and IoT devices. The simulator should :

1. Show the communication ability of these UAVs.

2. Allow the communication among UAVs in a device-to-device paradigm.

3. Show the energy consumption at different UAVs as a function of its mobility, sensors and IoT devices utilization.

4. Mimic the events happening in different regions.

Events may require different IoT devices and sensors. The information about different events would be forwarded to a centralized server for later processing and decision making.

1.2 Problem statement

The recent development of communication devices and wireless network technologies has had a great impact on the advancement and evolution of the Internet and telecommunications. The various "things", which include not only communication devices but also every other physical object on the planet, are also going to be connected to the Internet, and controlled through wireless networks. This concept, which is referred to as the "Internet of Things (IoT)", has attracted much attention of many researchers in recent years. The concept of IoT can be associated with multiple research areas. Unmanned Aerial Vehicle (UAV) network is one of the most recent emerging areas.

The name UAVs cover all vehicles, which are flying in the air with no person on-board with the capability of controlling the aircraft. UAVs became more attractive mainly with the rapid progress of circuitry and communication technologies. It is expected to be integrated into our society and support our daily life in the near future, principally with the interesting offered services that range from surveillance and bomb detection in military to journalism, shipping and delivery, rescue operations and health-care, disaster management in non military fields.

The various kinds of applications created using IoT technologies, more precisely the UAV networks, raise the need of designing an IoT simulator dedicated to reflect the UAV real environment before passing to its realization. This may show the eventual real effects of the imitated conditions in a less expensive fashion, and may also avoid a lot of real building problems. In addition, it allows the test of different designs before building one in hardware. Moreover, hypotheses about how or why certain phenomena occur can be tested for feasibility.

1.3 Project Framework

Our system aims to develop an Unmanned Aerial Vehicle and Internet of Things Simulator. We present in our solution a simulation of different scenarios using the UAVs. Passing through simulation is an essential step in the realization of any new solution. Indeed, simulation can be considered as a test tool that allows the implementation of various performance measures especially for systems whose evaluation is difficult to predict in the real case.

The main objective of our simulator is to test and evaluate different results of a UAV mission with different scenarios. In our case, this mission aims to collect data, calculate the autonomy which is the total time of the mission and the variation of energy in relation to the sensing and processing data, traveling, and communication between drone and base station. The detailed objectives will be presented in the next section.

1.3.1 Motivations of the project

The main motivations that led us to develop our system are:

- An increase in the need to design systems dedicated to the reflection of the real environment of the UAVs before proceeding to their realization, and to respond at the same time to the needs and demands of the UAV market.
- Provide designers with a platform to test different scenarios and enable them to make future decisions based on the simulation results before proceeding to hardware construction.
- Reduce costs at multiple levels and avoid many problems that may occur in the hardware deployment of the system.

1.3.2 Hypotheses

Before proceeding on the design part of the system, we consider the following assumptions:

1. The terrain is modeled in two dimensions (2D), so we do not take the altitude into account because the flights are made at constant elevation. Each point of the space will be presented by both latitude and longitude components only. For this, we considered the same presentation as Google Map.

2. We consider that the environment is barrier-free.

3. We do not take into account the avoidance of collisions between drones.

Our system can perfectly adapted to a case of application for smart cities, for collecting data in a sensor network, for monitoring a territory or for an intelligent traffic management system.

1.4 Research Objectives

The main objective of this project is to design an Internet of Things (IoT) simulator reflecting an Unmanned Aerial System (UAS) real environment, which is easy to use while being realistic. This will be achieved by targeting the following sub-objectives:

- The simulator will be a web application that represents a real Google Map. The later will be integrated using an existing script written on JavaScript and HTML. The web application will be developed by Django Rest framework which is dedicated to facilitate the creation of dynamic web applications, and python as a programming language.

- The considered Unmanned Aerial Vehicle (UAS) should include mobile Unmanned Aerial Vehicles (UAVs) dotted with different sensors that will be used to collect several data (such as temperature, humidity, photo, etc.) from the sensed area.

- These collected data would be showed to the user when required through the use of Representational State Transfer Application Programming Interface (REST API). The later facilitates the client-server communication.

- Data collection should be simulated using the web application interface, by allowing the user indicating the sensors' types, numbers, positions, and data collection times depending on drone's sensors.

- The simulator should support the addition and removal of sensors and IoT devices. This could be handled from the same previous web application interface by giving access to the user to manage several parameters of UAVs such as the position, altitude, speed, period of collect, energy, and the type and number of sensors included in each UAVs.

- The UAVs mobility should be integrated in a different server by using a Random Way Point mobility model or set a path according to the user. The server should be accessible from the web application interface through REST API.

- Taking into account the UAVs mobility, the simulator should give access to the user to select an area of flight or draw the desired path. It should also recover real time information (e.g., the current temperature, the current humidity, real photo from the satellite etc.) collected by each drone in the selected area of flight.

- It should also take into consideration (using existing mathematical models) the simultaneous run of several operations such as calculating the level of the drones' battery, their energy consumption, their lifetime, etc., and displaying them to the

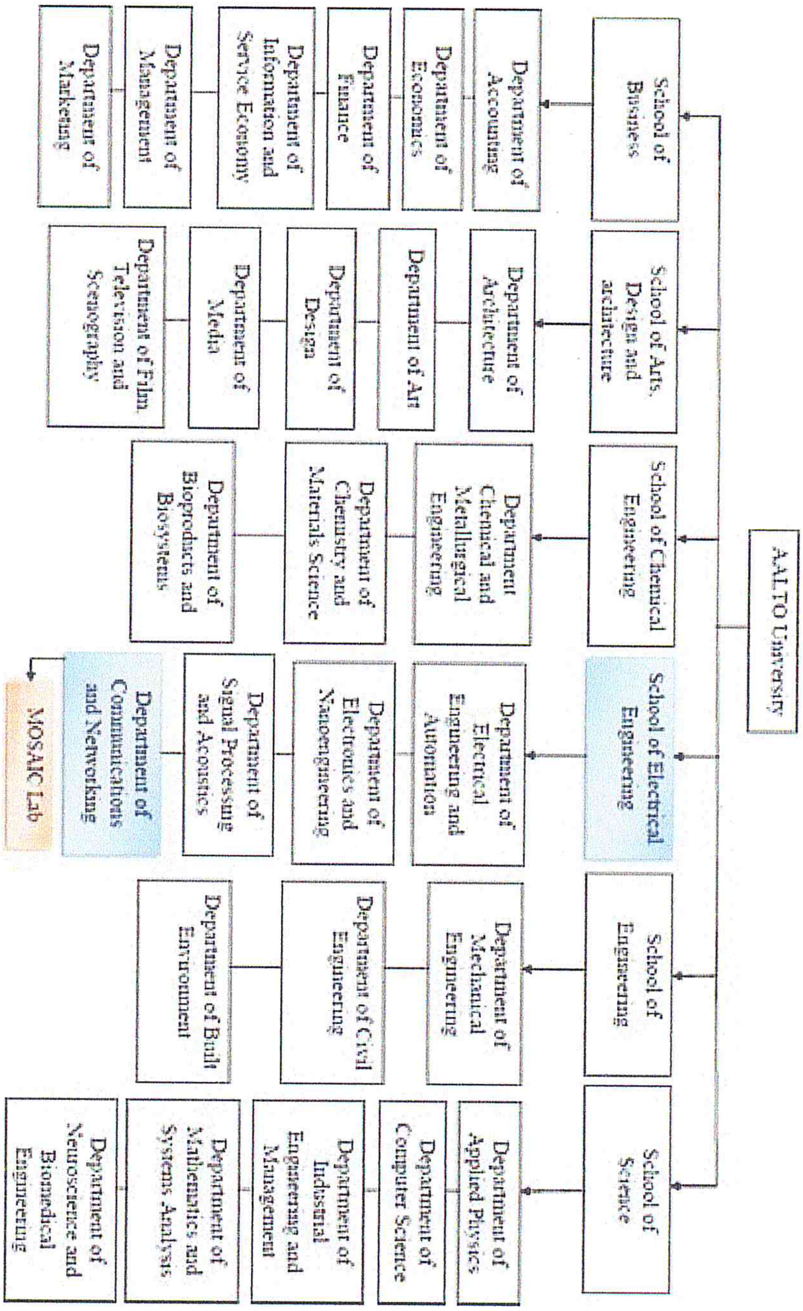


Figure 1: Organization Chart of Aalto University

2.1.3 Department of Communication and Networking ComNet

The Department of Communications and Networking belongs to the School of Electrical Engineering, Aalto University. It is a multi-disciplinary unit of research and higher education covering communications and networking technology, networking business, and human aspects of communication and communications technology [9].

In its area, ComNet is the largest unit in Finland. It develops communications, information and tele-traffic theory and conducts fundamental and applied experimental research in communications and networking technology.

2.1.4 MOSAIC Laboratory

The MOSAIC Lab is led by Prof. Tarik Taleb. The lab belongs to the Communications and Networking Department, School of Electrical Engineering, Aalto University. It consists of a group of highly enthusiastic researchers with strong expertise in different areas relevant to mobile networking and cloud computing [10].

The lab is involved in a number of research projects funded by Tekes, European Commission and EIT ICT Labs. It heavily works on Mobile Network Softwarization and Service Customization.

The main research activities are on the following three directions:

- Personalized Mobile Telecom based on big data and network function virtualization technologies.
- Drone-based platform for the delivery of Internet of Things services.
- Social network-based Personalized Over the top TV.

2.2 Research Center for Scientific and Technical Information CERIST

2.2.1 Presentation of CERIST

CERIST is an Algerian public organization with a scientific and technical characteristic. It was created in 1985 and it is now under the aegis of the Ministry of High Education and Scientific Research [11]. CERIST undertakes any research related to the creation, the setting up and the development of a national scientific and technical information system.

2.2.2 Organization Chart of CERIST

The figure 2 present the organization chart of the research center.

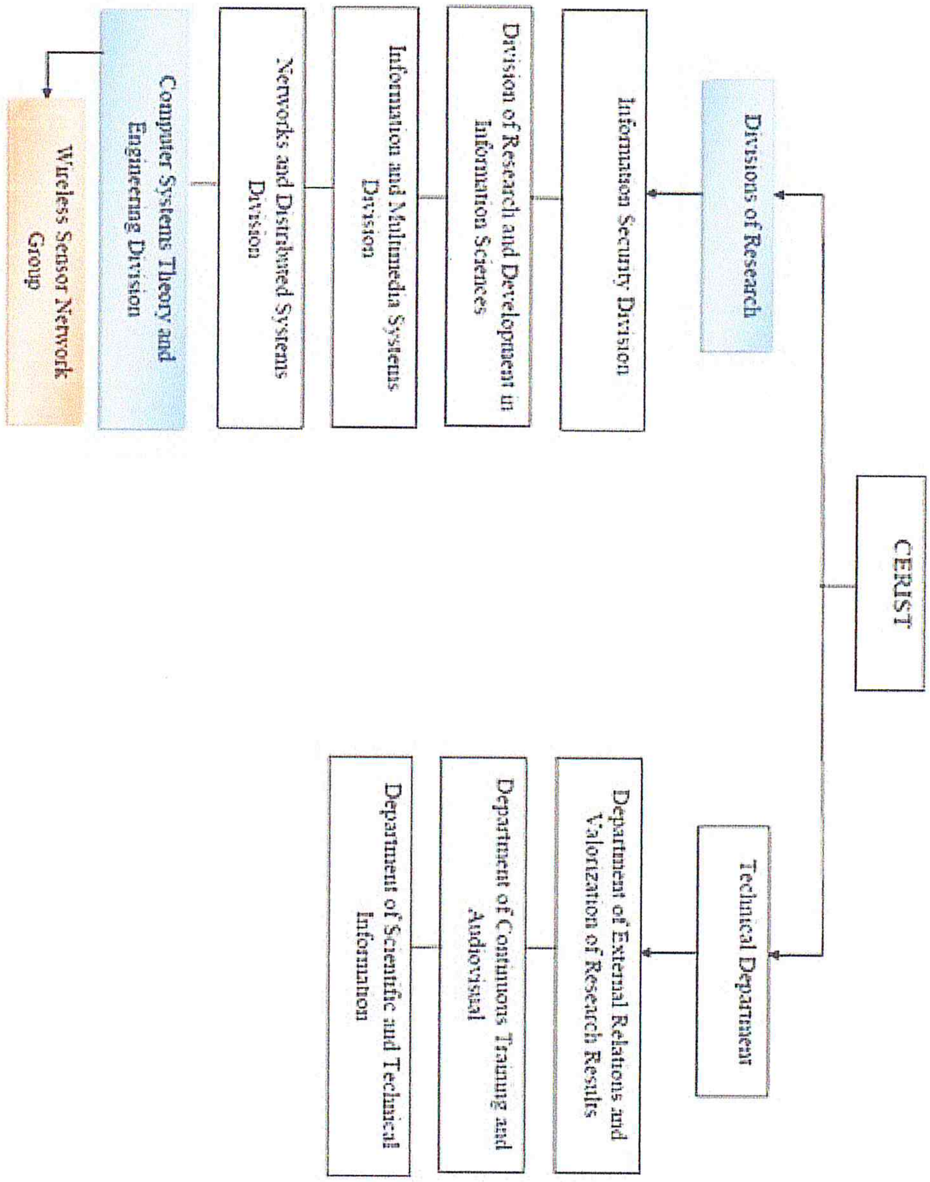


Figure 2: Organization Chart of CERIST

2.2.3 Computer Systems Theory and Engineering Division

The division Computer Systems Theory and Engineering explores the emergent problems related to the next generation infrastructures and distributed and pervasive systems. This division focus on the dynamic interactions and the wide scale resources sharing among the different communities and organizations combinating computing power, data, and human knowledge [12].

2.2.4 Wireless Sensor Networks Group

WSN Group a research group working on topics related wireless and sensor networking, Internet of Things applications. It is led by Prof. Djamel Djenouri within the Division of Computer Systems Theory and Engineering. Its research focuses on the design, analysis, and implementation of low-cost, scalable and efficient communication protocols, as well as real world applications [13]. Current research projects of the group are related to energy harvesting and green communications, energy control in smart buildings via IoT solutions and services.

3 Project Management

3.1 Structure of the document

The report is composed of four main chapters divided into two parts:

I. Part One: State of the Art:

This first part represents the state of the art on the studied subjects. It encompasses all the theoretical study we carried out in order to familiarize with the field and to acquire a theoretical background to run out the project.

1. The first chapter, entitled,**General information on Internet of Things, Unmanned Aerial Vehicles and simulation :**In this chapter, will serve to introduce the generalities of Internet of Things, Unmanned Aerial Vehicles and simulation, namely the different concepts, definitions and applications.
2. The second chapter, entitled, **Mission planning for Unmanned Aerial Vehicles:** In this chapter we will introduce the mission planning problem of an UAVs and we will give an overview of the similar research work done to solve this problem.

II. Part II: Proposed Solution:

This second part contains the proposed solution for the Unmanned Aerial Vehicles and Internet of Things simulator, the design and implementation of the system. It includes two chapters:

1. The third chapter, entitled,**Simulator Design:** In this chapter, we will describe the design of our simulator. This part will in particular comprise the various modules constituting our system.
2. The forth chapter, entitled,**Implementation and Tests:** In this chapter, we will describe the different parts of our system and the tests carried out to ensure that the system functions properly.

We will close this report with a general conclusion summarizing the main parts discussed and the perspectives drawn from this project.

Part One: State of the Art

Chapter 1

General Concepts on Internet of Things, Unmanned Aerial Vehicles and simulation

1.1 Introduction

In this first chapter, we introduce the Internet of Things and its various technologies, models of communication, as well as its challenges and application in different fields. We will then present concepts related Unmanned Aerial Vehicles, their components, and classes. Finally, we will present the concepts on simulation.

1.2 Internet of Things

The Internet of Things (IoTs) can be defined as connecting everyday objects like smartphones, Internet TVs, sensors and actuators to the Internet where the devices are intelligently linked together allowing new forms of communication between things and people, and between things themselves.

1.2.1 Definition

There is no single, universally accepted definition for the term Internet of Things. Different definitions are used by various groups to describe or promote a particular view of what IoT means and its most important attributes.

ITU-T [1] [14] defines Internet of Things (IoT) as a global infrastructure for the information society, enabling advanced services by interconnecting physical and virtual things based on existing and evolving inter-operable information and communication technologies.

1. ITU-T : International Telecommunication Union- Telecommunication Standardization Sector: a permanent organ of ITU. It's responsible for studying technical, operating and tariff questions and issuing Recommendation on them with a view to standardizing telecommunications on a worldwide basis.

It defines a 'thing' with regard to Internet of things as "an object of the physical world (physical things) or the information world (virtual things), which is capable of being identified and integrated into communication networks."

1.2.2 Basic Features

The fundamental characteristics of IoT are as summarized in Figure 3 [14], [15], [16], [17]:

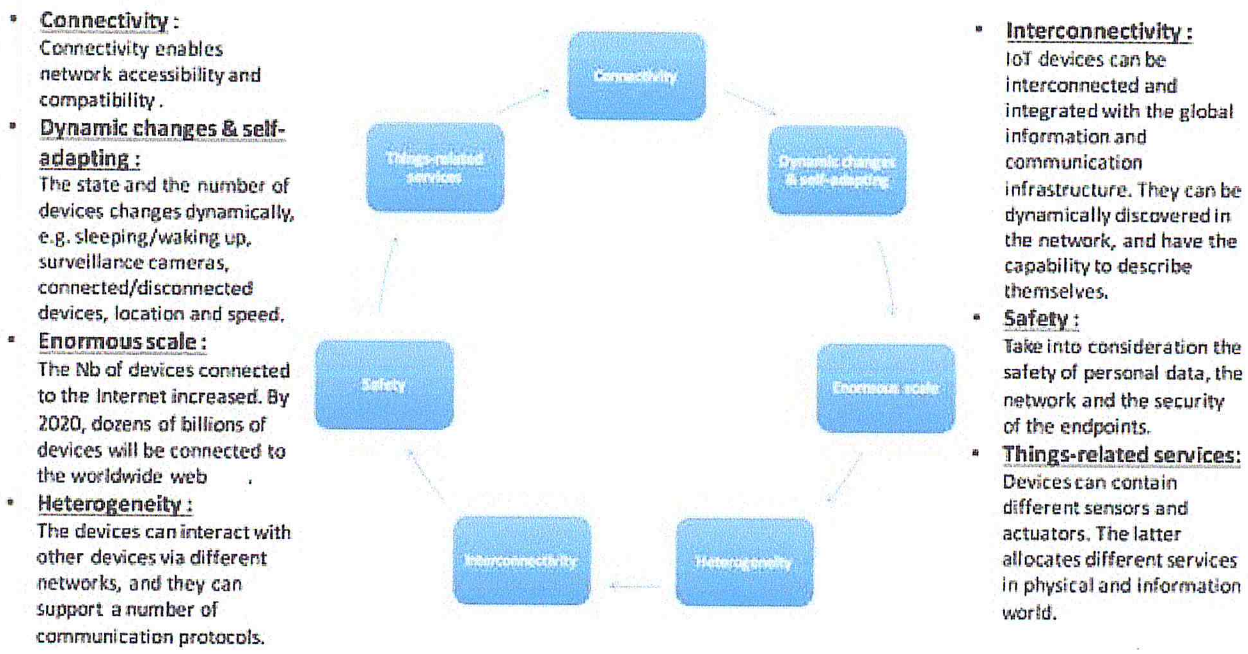


Figure 3: IoT fundamental characteristics

1.2.3 Internet of Things Technologies

IoT primarily exploits standard protocols and networking technologies. Wireless Network technologies presented in the figure 4 are the most frequently divided into four specific groups [18], [19]. The areas of application, the physical and medium access protocols used, and the signal range are the main criteria for this division [20].

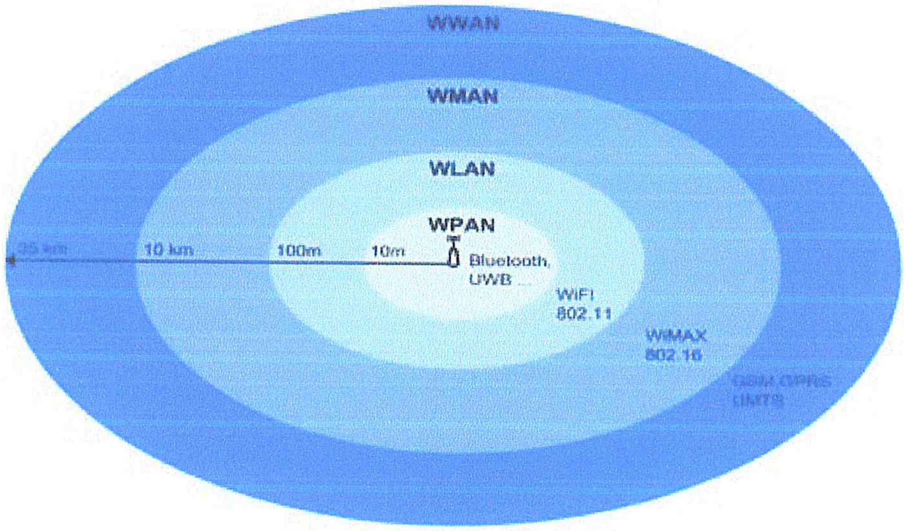


Figure 4: Wireless Network Divisions

1.2.3.1 Wireless Personal Area Network (WPAN)

It defines wireless networks that have a maximal signal range of 10 meters and these networks are used for the inter-connecting the respective devices one another.

a. Bluetooth

Bluetooth is a standard and a communication protocol with a low power consumption and designed for ranges between 10 to 100 m. It operates in 2.4 GHz frequency band and uses frequency hopping spread spectrum for the transmission [21].
The Bluetooth standard is divided into different standards, from IEEE.802.15.1 to IEEE 802.15.4.

The new Bluetooth Low-Energy (BLE) or Bluetooth Smart, as it is now branded – is a significant protocol for IoT applications [22]. Importantly, while it offers a similar range to Bluetooth it has been designed to offer significantly reduced power consumption.

b. Ultra-WideBand (UWB)

UWB (802.15.3) has recently attracted much attention as an indoor short-range high-speed wireless communication [23]. One of the most exciting characteristics of UWB is that its bandwidth is over 110 Mbps (up to 480 Mbps) which can satisfy most of the multimedia applications such as audio and video delivery in home networking and it can also act as a wireless cable replacement of high-speed serial bus such as USB 2.0 [21].

c. ZigBee

ZigBee (802.15.4) defines specifications for Low-Rate WPAN (LR-WPAN) for supporting simple devices that consume minimal power and typically operate in the Personal Operating Space (POS) of 10m. ZigBee provides self-organized, multi-hop, and reliable mesh networking with long battery lifetime [21], [24]. It is designed for reliable wirelessly networked monitoring and control networks.

1.2.3.2 Wireless Local Area Network (WLAN)

A WLAN is a wireless distribution method for two or more devices that use high-frequency radio waves and often include an access point to the Internet. It allows users to move around the coverage area, often a home or small office, while maintaining a network connection.

a. Wi-Fi
Wireless Fidelity (Wi-Fi) IEEE 802.11.a/b/g is a wireless technology that includes a set of standards for implementing WLAN communication in 2.4, 3.6, 5, and 60 GHz frequency bands [25]. Wi-Fi allows users to surf the Internet at broadband speeds when connected to an access point (AP) or in ad hoc mode. It is directed at computer-to-computer connections as an extension or substitution of cabled networks [21].

1.2.3.3 Wireless Metropolitan Area Network (WMAN)

The network working in accordance with this standard have a signal range of approximately 5 kms. It's used to connect the user to Internet. This standard is often called Worldwide Interoperability for Microwave Access (WiMAX).

a. WiMAW
WiMAX IEEE 802.16 is a wireless technology that aims at delivering broadband access over a large area. Depending on the frequency band, this technology applies frequency division duplex or time division duplex configurations. The data rate for the fixed standard supports up to 75 Mb/s (typically 20 to 30 Mb/s) per subscriber, while the mobile applications support 30 Mb/s (typically 3 to 5 Mb/s) per subscriber [26]. WiMAX is developed to handle high-quality voice, and video stream while providing high Quality of Services (QoS).

1.2.3.4 Wireless Wide Area Network (WWAN)

Wireless WAN is a wide area network in which separate areas of coverage or cells are connected wirelessly to provide service to a large geographic area such as cities or countries.

a. Global System for Mobile communication (GSM)
GSM is an open, digital cellular technology used for transmitting mobile voice and data services. GSM networks operate in four different frequency bands [27]. Most GSM networks operate in the 900 MHz or 1800 MHz bands. Some countries in the Americas (including the United States and Canada) use the 850 MHz and 1900 MHz bands because the 900 and 1800 MHz frequency bands were already allocated.

b. General Packet Radio Service (GPRS)
GPRS is an overlay extension for the GSM network to provide packet-based communication. It is designed to carry Internet Protocol (IP) and X.25 traffic destined to/from Terminal Equipment (TE) accessing the Wide Area Network (WAN) through a GSM wireless connection [28]. The bit rate of data transfer is not guaranteed because the GPRS works in the GSM networks, where the voice transfer has a higher priority.

c. Universal Mobile Telecommunication System (UTMS)

UMTS is a 3G cellular-system technology capable of providing multi-mobile services with multi-operators supporting a wide range. It is an umbrella term that encompasses the third generation (3G) radio technologies [29]. UMTS is designed to include not only traditional phone but also new technology tasks such as video calling, multimedia at data rates up to 2mbps, high-speed access to the world wide web-either directly on a handset or connected to a computer via Wi-Fi, Bluetooth.

1.2.3.5 Radio Frequency IDentification (RFID)

RFID is often seen as a prerequisite for the IoT [30]. A RFID system is consisted of tags (transmitters/responders) and readers (transmitters/receivers) [31]. The tag is a microchip connected with an antenna, which can be attached to an object as the identifier of the object. The RFID reader communicates with the RFID tag using radio waves. The main advantage of RFID technology is the automated identification and data capture that promises wholesale changes across a broad spectrum of business activities and aims to reduce the cost of the already used systems such as bar codes.

1.2.3.6 Near-Field Communication (NFC)

NFC, is a form of contactless communication between devices like tablets or smart phones. It is a derivative of radio frequency identification technology (RFID) in which one device sends information by touching another device [32]. It designed for low bandwidth (up to 424 kBaud) and short distance (up to 20 cm) communication. The main characteristic of NFC is that it is a wireless communication interface with a working distance limited to about 10 cm. The interface can operate in several modes. The modes are distinguished whether a device creates its own RF field or whether a device retrieves the power from the RF field generated by another device.

1.2.4 Internet of Things Communications Models

In March 2015, the Internet Architecture Board (IAB) released a guiding architectural document for networking of smart objects [33], which outlines a framework of four common communication models used by IoT devices.

1.2.4.1 Device-to-Device Communication Model

The device-to-device communication model represents a scheme, in which two or more devices are directly connected and communicate with one another without making use of any intermediary application server. These devices communicate over many types of networks, including IP networks or the Internet. They use protocols such as Bluetooth, Z-Wave, or ZigBee to establish direct device-to-device communications, as shown in Figure 5.

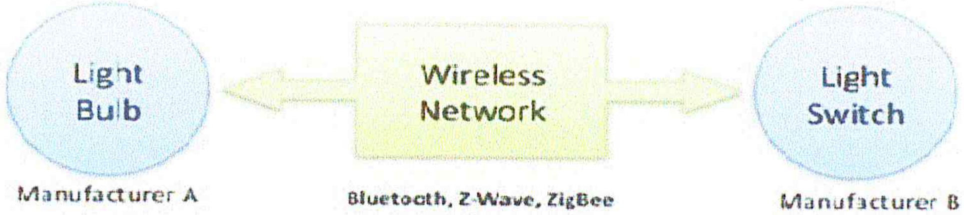


Figure 5: Device-to-Device communication model

This communication model is commonly used in applications such as home automation systems, which typically use small data packets of information to be communicated between devices with relatively low data rate requirements.

1.2.4.2 Device-to-Cloud Communication Model

In a device-to-cloud communication model, the IoT device connects directly to an Internet cloud service to exchange data and control message traffic. This approach frequently takes advantage of existing communication mechanisms including traditional wired Ethernet or Wi-Fi connections to establish a connection between the device and the IP network, which ultimately connects to the cloud service. This is shown in Figure 6.

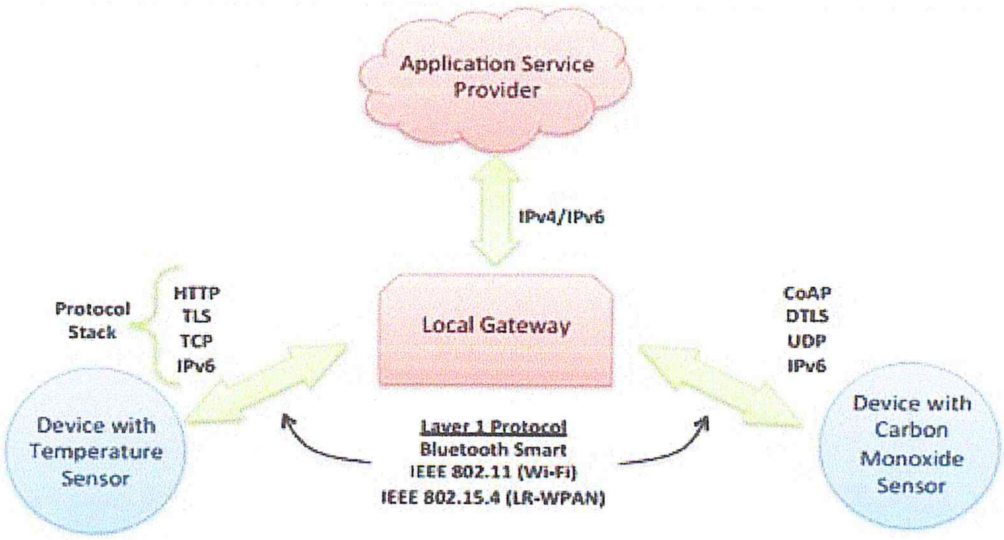


Figure 6: Device-to-Cloud communication model

This communication model is employed by some popular consumers of IoT devices such as the Nest Labs Learning Thermostat [34] and the Samsung Smart TV [35].

1.2.4.3 Device-to-Gateway Communication Model

In the device-to-gateway model, or more typically, the device-to-Application-Layer-Gateway (ALG) model, the IoT device connects to local gateway which is connected to the cloud services. Here, local gateway is considered as an intermediate server between devices to the cloud. The model is shown in Figure 7.

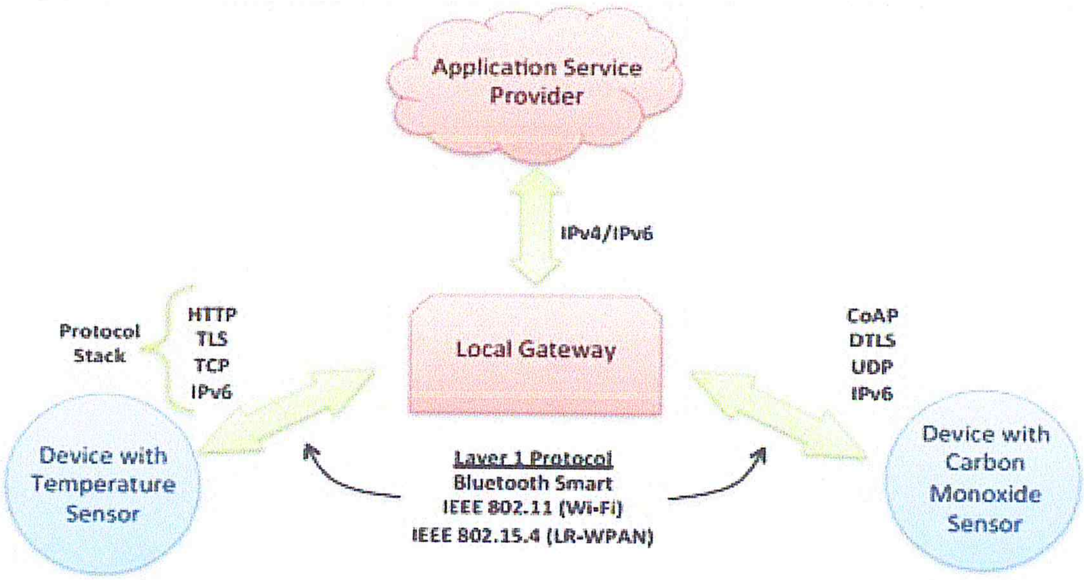


Figure 7: Device-to-Gateway communication model

Local gateway is usually a smart phone, which provides security and other functionality such as relay data to a cloud service, or protocol translation.

1.2.4.4 Back-End Data-Sharing Communication Model

The back-end data-sharing model refers to a communication architecture that enables users to export and analyze smart object data from a cloud service in combination with data from other sources. In this architecture, the device is connected directly to only one provider and that provider is getting data from other source in the back-end. A graphical representation of this design is shown in Figure 8.

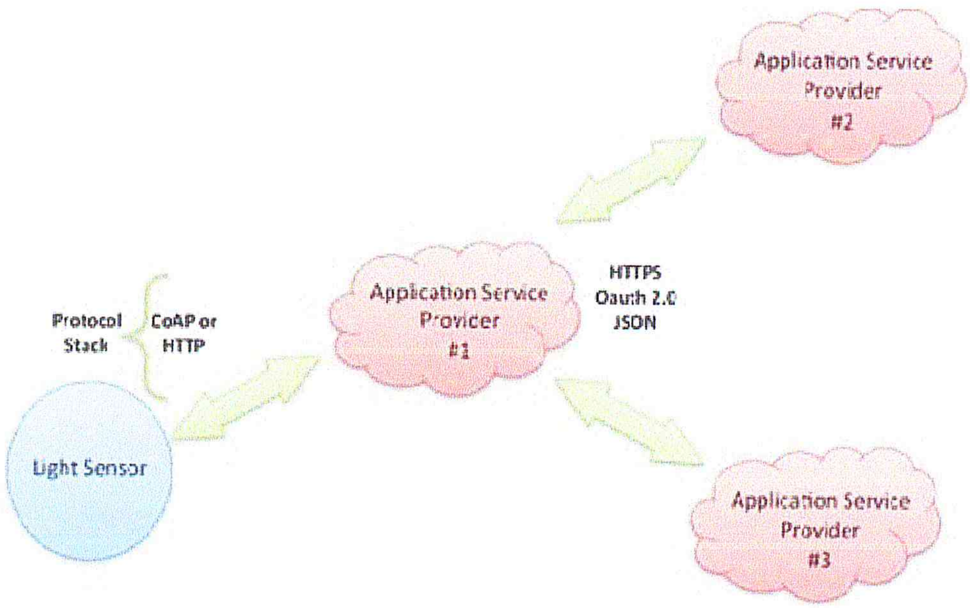


Figure 8: Back-End Data Sharing communication model

The back-end data-sharing model suggests a federated cloud services approach or cloud Applications Programmer Interfaces (APIs), which are needed to achieve interoperability of smart device data hosted in the cloud.

1.2.5 Internet of Things application domains and use cases

The potentialities offered by the IoT make it possible to develop numerous applications based on it. However, only few ones are deployed currently. As shown in Fig. 10, the major application fields for the IoT are the creation of smart environments (spaces) and self-aware things (e.g., smart transport, smart buildings, and in general smart cities, etc.) that will be activated to realize entertainment and sports, logistic, agriculture, education, governmental services, food, energy, mobility, digital society and e-health applications [36], [37], [38], [39], [40], [41]. In Figure 9, some applications are presented:

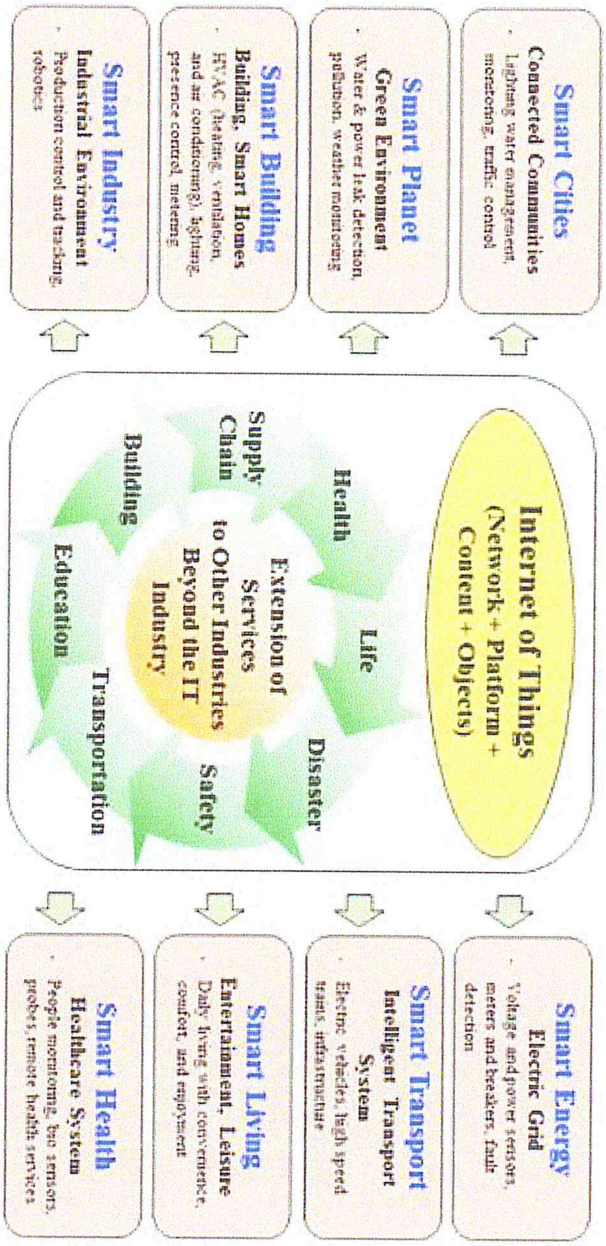


Figure 9: Internet of Things examples

1.2.6 Challenges

IoT applications and scenarios outlined above are very interesting and provide technologies that allow everything to become smart. Nevertheless, there are some challenges related to IoT application concept [42], [43], [44]. Figure 10 illustrates these challenges.

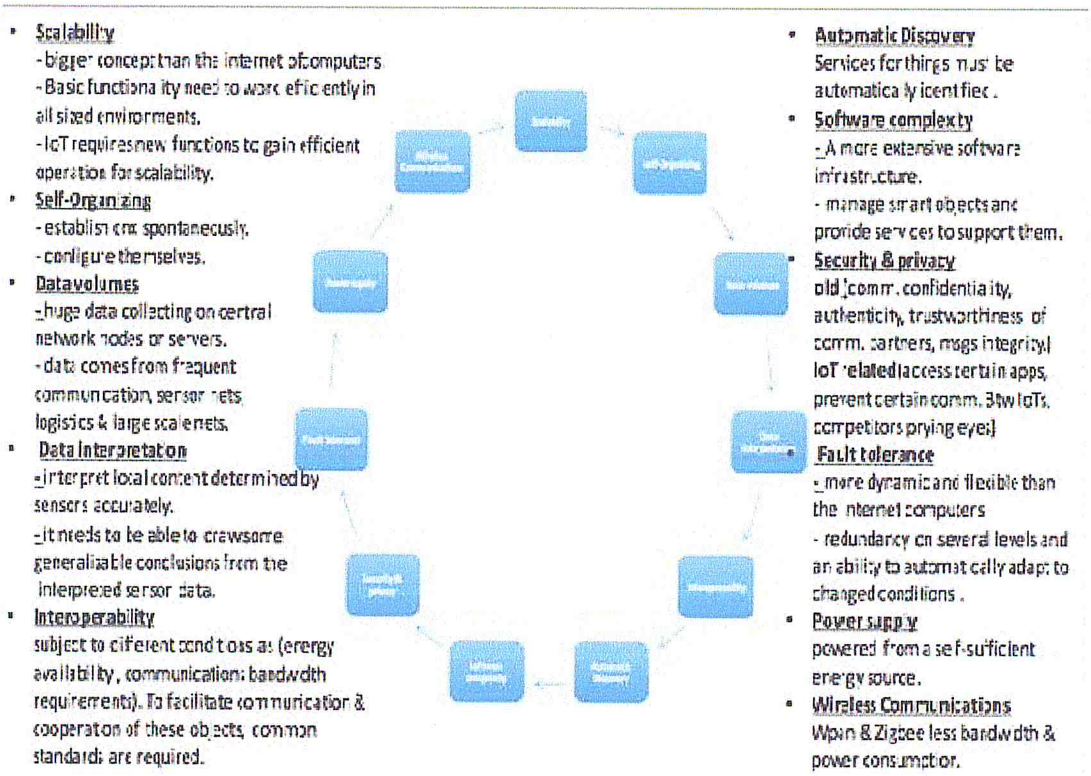


Figure 10: IoT challenges

1.2.7 Internet of Things benefits

There are many advantages of incorporating IoT into our life, which can help individuals, businesses, and society on a daily basis [45], [46]. Some of the most important benefits of IoT are the following:

1.2.7.1 Assessing Web User Intelligence

IoT is used by third party web data aggregators to have a better understanding of their key customer by tracking them on social media networks to know their preferences.

1.2.7.2 Enhancing Data Collection

Modern data collection suffers from its limitations and its design for passive use. IoT breaks it out of those spaces and places it where humans really want to go to analyze our world. It allows an accurate picture of everything.

1.2.7.3 Integration into Health Care System

This could prove to be incredibly beneficial for both individuals and society. A chip could be implemented into each individual, allowing for hospitals to monitor the vital signs of the patient.

1.2.7.4 Inventory Management

IoT is used to tag Radio Frequency sensors to track the location of products in real time. It has been instrumental in tracking the level of inventory and to stock it in advance, making alerts for unforeseen stoppages, automatically placing orders, etc.

1.2.7.5 Reduce Waste

IoT makes areas of improvement clear. Current analytics give us superficial insight, but IoT provides real-world information leading to more effective management of resources.

1.2.7.6 Technology Optimization

The same technologies and data which improve the customer experience also improve device use, and help in more potent improvements to technology. IoT unlocks a world of critical functional and field data.

1.2.7.7 Transportation

IoT eases and simplifies the entire process by introducing a monitory sensor that helps to track distance and time locations, and other contributing factors.

1.2.7.8 Risks

Although IoT provides an impressive set of benefits, it presents a significant set of risks [45], [46]. Some major issues are the following:

1.2.7.9 Complexity

Some find IoT systems complicated in terms of design, deployment, and maintenance given their use of multiple technologies and a large set of new enabling technologies.

1.2.7.10 Compliance

IoT, like any other technology in the realm of business, must comply with regulations. Its complexity makes the issue of compliance seem incredibly challenging.

1.2.7.11 Flexibility

Many are concerned about the flexibility of an IoT system to integrate easily between each other. They worry about finding themselves with several conflicting or locked systems.

1.2.7.12 Privacy

The sophistication of IoT provides substantial personal data in extreme detail without the user's active participation.

1.2.7.13 Security

IoT creates an ecosystem of constantly connected devices communicating over networks. The system offers little control despite any security measures. This lets users be exposed to various kinds of attackers.

1.3 Unmanned Aerial Vehicles

UAVs, also called unmanned aircraft systems (UAS) [47] or drone, have recently reached unprecedented levels of growth in diverse military and civilian application domains to accomplish difficult tasks in very hostile environments, without any risk to humans.

1.3.1 Definition

The name UAV covers all vehicles, which are flying in the air with no person onboard with the capability of controlling the aircraft. This term is used commonly in the computer science and artificial intelligence community, but other terms exist, such as [48]: Remotely Piloted Vehicle (RPV), Remotely Operated Aircraft (ROA), Remote Controlled Helicopter (RC-Helicopter), Unmanned Vehicle Systems (UVS). Drone and model helicopter are often used, too.

UAV can be remotely controlled by a pilot at a ground control station or can fly autonomously based on pre-programmed flight plans or more complex dynamic automation systems. It may carry a lethal or non- lethal payload.

1.3.2 Components

The UAV is in fact only one element of a system, designed and deployed to carry out one or more missions [49]. This is why specialists talk about UAVs systems or UAS. The figure 11 demonstrates the main components of UAV:

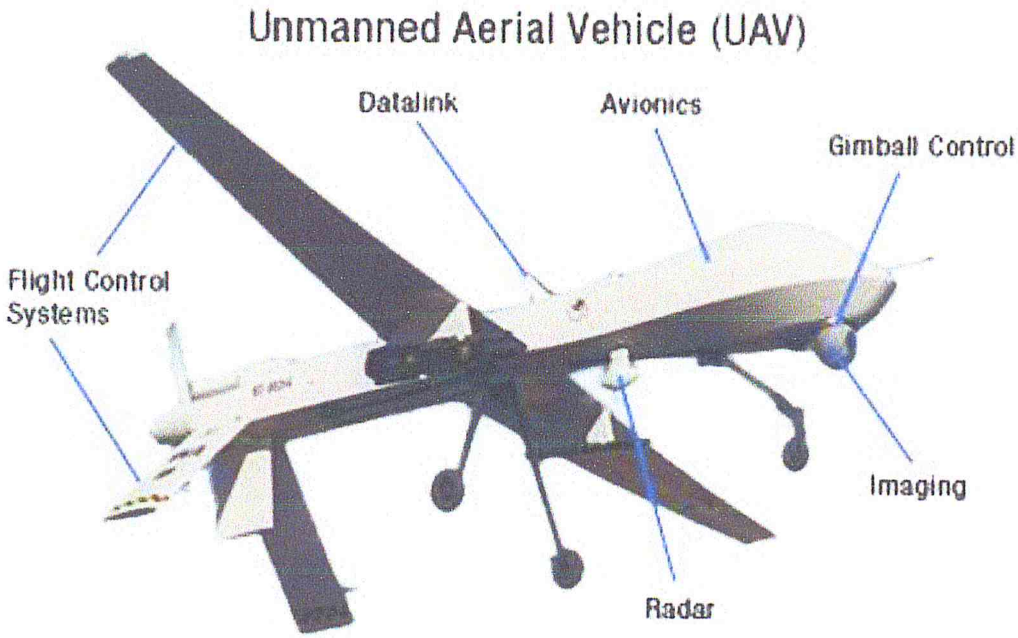


Figure 11: Unmanned Aerial Vehicles

A UAS describes the entire system that includes aircraft, control stations, and data links [50] [51] [52]. The main components of the UAV are the following [53], [54]:

1.3.2.1 Airborne part – aircraft

- Airframe – wings and body.
- Engine – most UAVs are powered by a piston (reciprocating) engine driving a propeller.
- Sensors – radar, photo or video camera, IR scanners or ELINT are most common. Sensors may include a (laser) target designator to provide guidance for stand-off guided missiles and shells.
- Control system – used to fly the UAV. Either a two-way data link (radio) for remote control or an on-board computer (generally with GPS navigation) connected to the aircraft control system.
- Data link – One-way (radio) link transmitting data collected by sensors.
- Recovery system – optional.

1.3.2.2 Ground-based part

- Launcher – many UAVs are launched by a catapult type launcher or with a rocket booster.
- Transport and maintenance – UAV systems are generally mobile.

- Control system – used to fly the UAV; this includes a "cockpit" from which the "pilot" on the ground flies the UAVs (if remote-controlled). It is linked by a two-way (radio) link to the UAV. The control system may include sub-control systems allowing other operators to take over flight.
- Data link – receiver for sensor data transmitted from the UAV. There may be several receivers of data, including some not part of the UAV system.

1.3.3 Classifications

Several groups have proposed the creation of reference standards for the international UAV platform community. The European Association of Unmanned Vehicles Systems (EURO UVS) has drawn up a classification of UAV systems, based on parameters such as flight altitude, endurance, speed, maximum take of weight (MTOW), size, and so forth [55].

1.3.3.1 Micro and Mini UAVs

Micro and mini unmanned aerial vehicle are the smallest UAV technology. These drones can fly at low altitudes (<300 m), and they are suitable to interact with buildings. These micro and mini UAVs can carry little and lighter things such as listening and recording devices. Transmitters and cameras can also be carried by mini UAVs. Micro UAVs are smaller as compared to mini UAVs. The weight of the mini UAVs are less than 30 kilograms and are appropriate for commercial applications.

1.3.3.2 Tactical UAVs

Tactical UAVs are heavier as compared to mini and micro UAVs. These UAVs can fly at higher altitudes. They are best for military applications. Tactical UAVs have further categories like close, short, medium, long, endurance and altitude long range. These use more advanced technology.

1.3.3.3 Strategic UAVs

Strategic UAVs can fly with high speed at higher altitudes and distances. These are the heavies UAVs. These UAVs have maximum take-off weight and they can carry heavy payloads. Sophisticated equipment can also be carried by strategic drones. These UAVs are used for mapping, surveillance, and photography.

1.3.4 Unmanned Aerial Vehicles uses

Advances in technology with a constant drop in costs have led to an increased use of unmanned aerial vehicles by the military and civilian sectors [56]. UAVs have found a number of applications in the military and defense world to avoid being detected and destroyed by the enemy [57]. They are used for the surveillance in order to ensure protection of the people and places. These unmanned aerial vehicles are also used for the purpose of air strikes [58] and in BOMB detection [59].

Civilian drones have become an inexpensive and accessible way to help accomplish many tasks [57], [60], [61]. They are used for disaster management by assisting people, reducing the number of victims and avoiding the economic consequences [62]. They are also used for shipping and delivery, take aerial photography, geographic mapping, archaeological surveys, agriculture, weather forecasting, rescue operations and health-care, etc.

1.3.5 Challenges

The development of Unmanned Aerial Systems (UAS) offers benefits in a wide variety of applications, to realize it, a number of challenges need to be addressed by the research community, such as [63], [64]:

- Regulation of the airspace: the access to the airspace must be well organized and controlled.
- Protection of the privacy and safety of people.
- The spectrum regulation: the communication of UAVs should not affect the communication of current air traffic.
- Resource management.
- Selection of an appropriate wireless technology such as cellular, WiMAX, Wi-Fi, etc. depends on the UAV task type, its duration, and the operation environment, e.g. LTE-m for the delivery of sensed small data.
- Path planning/mobility.
- Energy efficiency.
- Channel modeling.
- Handover and moving cells.
- Interference management.
- The send-and-receive signal strength (RSSI): which is principally due to the dynamic and high mobility feature of UAVs.
- Enhancing the mobile core network for the efficient support of UAV-based mobile communications.

1.3.6 Advantages

UAVs may be employed for a wide range of transportation operations and planning applications. Incident response, monitor freeway conditions, coordination among a network of traffic signals, traveler information, emergency vehicle guidance, track vehicle movements in an intersection, measurement of typical roadway usage, monitor parking lot utilization, estimate Origin-Destination (OD) flows [65]. Some of the advantages of UAVs are:

- They can move at higher speeds than ground vehicles as they are not restricted to traveling on the road network.
- They have advantages over manned vehicles as most of the functions and operations can be implemented at a much lower cost, faster and safer.
- UAVs may potentially fly in conditions that are too dangerous for a manned aircraft, such as evacuation conditions, or very bad weather conditions.
- UAVs are programmed off-line (when Global Control System is lost) and controlled in real-time to navigate and to collect transportation surveillance data.
- UAVs can view a whole set of network of roads at a time and inform the base station of emergency or accidental sites. It also permits timely view of disaster area to access severity of damage. The base station can then choose the best route and inform the police cars.

1.3.7 Disadvantages

Unmanned Aerial Vehicles are relatively expensive to produce (the costs may go down over time though). Further, a human mistake in the remote controls can cause the plane to crash. The computer systems or the software could break down resulting in loss of the plane which costs millions, and/or casualties on the ground [63]. The computer malfunction can take place resulting in the loss of control in the aircraft. There is the ethical question on the use of the autonomous system in the combat situations especially as the computer cannot take the initiative, which can result in the civilian deaths [66].

The psychological issue has become apparent in the military used of Unmanned Aerial Vehicles due to the personnel being able to click the button and kill numerous people and then go home [67], without being in the normal war situation, so the drones or the fleet of drones can be taken and controlled by the enemy.

1.4 Simulation

1.4.1 Definition

Simulation is the limitation of the operation of a real-world process or system over time [68]. It is a particular type of modeling which has "inputs", and "outputs" which are observed as the simulation runs [69]. Frequently, the inputs are the attributes needed to make the model match up with some specific social setting and the outputs are the behaviors of the model through time.

1.4.2 Simulation uses

Simulation is used in many contexts, such as performance optimization, safety engineering, testing, training, education, and video games. It's also used to [69], [70]:

- Make decisions on very complex problems for which there are no optimal solutions.
- Solve problems which need analytical (logical reasoning) approaches which can't be definitively quantified.
- Attempt straight and optimal solutions and decisions.
- When it is not advisable to experiment with reality itself.
- Study almost any problem that involves uncertainly.
- Where mathematical simplification is not feasible.
- When real system can get very costly.

The integration of new technologies with UAVs requires simulation before the launch of new systems. The main added value of this simulator of drones is not to risk to damage a real drone. This saves money and equipment. On the other hand, it is quite possible to create fault simulations, for example critical failures (engine loss, control, the risk of the drone falling). This allows the developer to train in the risks, and be able to act accordingly, thus avoiding costly high errors of prototyping and testing and reflecting the actual environment.

1.4.3 Examples of simulator

Simulation is used in different fields such as computer science, health-care, entertainment, manufacturing, flight, etc. There is several software simulators developed in several domains like [71], [72], [73], [74], [75]:

- Galatea – a multi-agent, multi-programming, simulation platform.
- SageMath – a system for algebra and geometry experimentation via Python.
- SimPy – an open-source discrete-event simulation package based on Python.
- Tortuga – an open-source software framework for discrete-event simulation in Java.
- FlightGear – a 3D open source simulator, very realistic and focused on simulating the flight of a single aircraft vehicle.
- Aerofly RC 7 – an UAV radio controlled flight simulator.
- DWR – a LightWeight multi-UAV simulator.
- Simdrone – a real-time UAV flight and payload simulator.
- ns-3 – an open source network simulator.
- Mission Planner – a full-featured ground station application for the ArduPilot open source autopilot project.
- FPV PASSION – an UAV simulator based on Google Earth.
- RDS – a racing drone simulator.

1.4.4 Advantages

Simulation is best suited to analyze complex and large practical problems when it is not possible to solve them through a mathematical methods and models. Some of the most important advantages of simulation are the following [70], [76]:

- Straight forward and flexible.
- Suitable to analyze large and complex real-life problems.
- Sometimes simulation is the only method available.
- It may be used over and over to analyze different situations.
- Breaking down of complicated systems into sub systems.
- Data for further analysis can easily be generated.
- Avoids the cost of real world experimentation.

A flight simulator or UAV simulator allows to train to the piloting of drones by doing exercises playful and evolutionary. It also allows to measure the performances and to test the feasibility of the different scenarios of a drone mission before proceeding to the realization. It also helps the developers to make future decisions according to the results obtained.

1.4.5 Disadvantages

There are a number of drawbacks using simulations that must not be ignored [70], [76].

- Simulation models are expensive and take a long time to develop.
- It is a trial and error approach that many produce different solutions in repeated runs.
- It is often too long and a complicated process to develop a model.
- Simulation results are sometimes hard to interpret.
- Difficult for people to understand that they are not looking at reality but an abstraction of the real world.
- A large amount of time may be required to develop the simulation.
- The simulation model does not produce any answers by itself, the user has to provide all the constraints for the simulations that he wants to examine.

1.5 Conclusion

Unmanned Aerial Vehicles are rapidly expanding their capabilities beyond military and civilian applications. They play a role in the Internet of Things because they are critically dependent on sensors, antennas and embedded software to provide two-way communications for remote control and monitoring. Drone simulation will, therefore, be critical in achieving estimates and reflecting the real environment.

In this chapter, we have presented various characteristic of IoT as well as its communication technology. The concept of Unmanned Aerial Vehicles was presented and a classification was given relative to a certain parameter like flight altitude, endurance, and speed, etc. Finally, the simulation has been introduced as a soft concept, which we will use in our work.

Chapter 2

Mission Planning for Unmanned Aerial Vehicles

2.1 Introduction

Recent technological advances enabled the development of UAVs, which have shown great potential in both military and civilian applications. However, the full potential of these aerial vehicles would be complete only when they can operate autonomously. To achieve this autonomy, several challenges must faced, such as the planning the mission of a fleet of autonomous drones in areas of obstacles and threats.

The mission must be carried out in such a way as to fulfill the tasks entrusted without compromising the success of the mission. The design and optimization of autonomous trajectories for these vehicles is a complex and crucial problem in mission planning and control. This topic has been attracting researchers in the fields of robotics, automatic but also computer science and many others.

In this chapter, we first introduce the vocabulary related to mission planning as well as the related constraints. In the second part, we will present some models of the path planning problem, and for the problem of vehicle traffic which has its application in the case of UAVs. We briefly present the various research studies carried out in this context. We will end this chapter with a conclusion followed by a summary of research summarizing the works presented.

2.2 Mission of Unmanned Aerial Vehicles

2.2.1 Mission Planning

2.2.2 Planning

Mission planning can be described as a set of objectives that must be met within a set of constraints defined by resource limitations. It has been a field of research in Artificial Intelligence (AI) for over three decades. Various areas, including robotics, web information gathering, and mission control have benefited from planning techniques., and several research projects are being carried out in this field [77].

2.2.2.1 Autonomous system

An autonomous system is a systems that is aware of/ and interact with Its environment. That is, a system that can be programmed to automatically carry out specific operational missions (with a certain level of self governance), under the control and guidance of a human-operator at mission level [78].

2.2.3 Mission of Unmanned Aerial Vehicles

Mission planning for UAV can be defined as the planning process of the locations to visit (target or waypoints) and the actions to perform (loading/dropping a load, taking videos/pictures, acquiring information), typically over a time period. Functionally, mission planning resides above the process of path planning, where the mission planner generates the desired mission plan, and then the path planner generates the flight plan (trajectories) between the waypoints.

Each target is defined by its geographic coordinates, and by the angle in which the target is viewed. The temporal window is also part of the elements defining a target if the target must be visited in a well defined time interval [79].

The problem of mission planning for a fleet of UAVs has three sub problems:

1. A problem of path planning to reach the targets.
2. A problem in allocating targets to drones.
3. A problem of optimization of trajectories, which aims to verify that the defined roads can be borrowed by the vehicles in real conditions, i.e, taking into account the navigation constraints and the kinematics of drones.

An essential concept in planning a mission is cooperation or collaboration. Cooperation occurs at a higher level when different UAVs work together in a common mission [78].

2.2.3.1 Path planning

Path planning provides a set of waypoints defining the routes that the UAVs can take to reach their targets, while considering obstacles and threats. The length of the paths obtained will be used for the assignment of the targets to the vehicles, as well as the probability of survival of the vehicle are calculated through the risk and the dangers of the road. Threats from enemy radars in the case of a military mission are also criteria to be taken into account [80], [81], [82].
The expected result through this step is to establish an array containing all the paths from the source to the destination (target) with their respective costs. It can be assimilated to a routing table.

2.2.3.2 Target allocation to unmanned aerial vehicles

It consists of assigning tasks (visiting a target, taking a photo, etc.) to the different devices. It is a constraint optimization problem whose goal is to minimize a cost function or to maximize a reward function [83]. In the case of a UAV mission, the distance traveled must be minimized in order to minimize the total mission completion time and energy consumption of the UAVs, to maximize the survival time of the UAVs, as well as the reward (R), which is the sum of the rewards obtained from the targets.
A modeling of these constraints is presented in [84]. The used notations are summarized in Table 1.
We name by:

Notation / variable	Denomination
N_t	Number of targets
N_v	Number of UAVs designed for the mission
r_k	Reward associated with target k
x_k	Decision variable about the processing of the target k
d_i	Distance traveled by the vehicle, i
P_i	Probability of survival of the vehicle, i

Table I: Notations and variables used for the mathematical modeling of the constraints of the target allocation model for UAVs

Thus, the objectives are:

$$\begin{cases} MaxR = \sum_{k=1}^{Nt} x_k.r_k \\ minD = \sum_{i=1}^{Nv} d_i \\ MaxT = \sum_{i=1}^{Nv} p_i \end{cases} \tag{1}$$

The aim of this multi-objective problem is to maximize the reward associated with the mission that can be defined by several criteria, the probability of survival of the vehicles, and minimizing the duration of the mission. To decide about the optimality of a solution for the problem of mission planning, the three objectives mentioned above have to be combined.

2.2.3.3 Calculation of trajectories

From the selected points which define the raw paths of the drones, the objective is to refine these paths to generate trajectories as a function of time that satisfy the kinematic constraints of the vehicles.
The authors in [85] and [86] propose optimization methods based on the positioning of circles at the points of passage requiring a change of direction and of a radius. The latter equals to the radius of gyration of the vehicles at a constant fixed speed.

2.3 Constraints of a UAV mission

The constraints to be respected for a UAVs mission and during the modeling can be summarized in the following points:

- Ensure visitation of known, fixed, or moving targets.
- Minimize the duration of the mission.
- Maximize the number of targets visited by each UAV.
- To compensate for the failure or the loss of part of the drone fleet when it leads to mission failures.
- Collect data.
- Minimize the number of calls.
- The safety of UAVs during the mission.
- The respect of the kinematic [1] constraints and maneuverability of the vehicles.
- The reactivity of the strategy to be used, since the context is dynamic and therefore requires real-time operability.

Route planning imposes further requirements in addition to those previously mentioned, such as the coordination of the various UAVs. The UAVs share the same space and aim to achieve the same mission, which requires constant communication.

2.4 Problem modeling

Mission Planning Problem (MPP) is one of the problems of combinatorial optimization NP-Hard Problem [87], [88]. Therefore, for resolution of this problem on within a polynomial time, only non-deterministic algorithm can be used for the moment.
This problem has attracted the interest of several researchers and has been the subject of several works.

In the remainder of this chapter, we will present the main modeling proposed in the literature for the planning of a drone fleet mission.

2.5 Revue of the literature on unmanned aerial vehicles path planning problems

UAVs are widely used in military and civilian applications, including in surveillance and fire fighting. Routing problems in these applications can be categorized into different variants of the Vehicle Routing Problem (VRP), which has broad applicability in the areas of transport, distribution and logistics [89].

Trajectory planning can be categorized into variants of the vehicle routing problem, which is also a variant of the VRP. The aim is therefore to determine the optimal set of routes to be covered by a fleet of vehicles that might bet serving customers, or just to collect data.

1. Kinematics is the most basic study of how mechanical systems behave.

2.5.1 Travelling Salesman Problem (TSP)

The task of TSP is to arrange a tour of n cities such that each city is visited only once and the length of the tour (or some other cost function) is minimized [90]. For an exact solution, the only known algorithms require the number of steps to grow at least exponentially with the number of elements in the problem. The main difficulty of this problem is the immense number of possible tours: (n-1)!/2 for n cities [91].

The mathematical modeling of the problem takes the form of a graph where each city is represented by a node and the arcs, that might be oriented or not, to connect each pair of nodes. Each arc is weighted proportionally to the distance between its two extremities. When the trade traveler can go to all cities from all the others, the problem is said to be complete. Linear modeling to solve this problem was proposed for the first time in 1954.

Table 2 summarizes notations and variables used in the remaining of the chapter.

Notation / variable	Denomination
i, j	Node to visit
N	The list of all nodes
k	UAVs
c_{ij}	The cost relative to the arc (i,j) (the distance)
x_{ij}	Binary variable, equal to 1 if (i,j) is part of the path. (Case of PVC)
t_{ij}	The time required for the arc path (i,j)
$[a_i, b_i]$	Time window of i.
${x_{ij}}^k$	Binary variable, equal to 1 if (i, j) is part of the path of the UAV k.
${y_i}^k$	Binary variable, equal to 1 if the vehicle k crosses the node i.
${u_i}^k$	The time of arrival of the vehicle k at the node i.
s_i	The duration of service at node i.
S	Sub tour

Table II: Notations and Variables used for the mathematical modeling of TSP and VRP.

2.5.1.1 Mathematical formulation of the TSP

The authors in [92] define the problem as follows:

$$MinZ = \sum_{i \in N} \sum_{j \in N} x_{ij} c_{ij} \tag{2}$$

Under the constraints:

$$\sum_{i \in N} x_{ij} = 1, \qquad \forall j \in N \tag{3}$$

$$\sum_{j \in N} x_{ij} = 1, \qquad \forall i \in N \tag{4}$$

$$\sum_{ij \in S} x_{ij} \leq S - 1, \qquad \forall S \subset N; \qquad 2 \leq |S| \leq n - 2. \tag{5}$$

$$x_{ij} \in \{0, 1\} \qquad \forall j, j \in N. \tag{6}$$

The objective is to minimize the total distance traveled during the tour. Constraints (3) and (4) ensure that each vertex is visited only once. (5) eliminates sub-turns within a turn that must start and end at the same point.

| S | : The number of vertices belonging to a potential sub-turn composed of the set S. The last constraint (6) ensures that the variables are binary.

The Travelling Salesman Problem (TSP) is a well known NP hard problem. The traveler will be the vehicle and the cities will be the targets to visit. Other variants of this problem have also been used to model aircraft flight path planning. In the following we will present the most used variants in the domain of UAVs.

2.5.1.2 Solution of Travelling Salesman Problem

There are two solutions of TSP. The first one will find the optimal solution which is close to the exact solution. This will guarantee the quality of the solution, but it is takes a long time. The second will find the nearest optimal solution within a reasonable time. It will not guarantee that the solution is closeness to the exact solution. To improve the solution space and to increase the performance, genetic algorithms are used to solve the travelling salesman within a reasonable amount of time [93].

2.5.2 Vehicle Routing Problem VRP

Vehicle Routing Problems are basically concerned with finding a way to visit a given set of customer locations using a given set of vehicles in such a way that a cost function, often the total distance, is minimized [94].
In the most basic version of this problem, each customer must be visited by exactly one vehicle and each vehicle performs one trip, starting and ending at a depot location.

2.5.2.1 Mathematical formulation of the VRP

The objective function:

$$MinZ = \sum_{k\in K}\sum_{i\in N}\sum_{j\in N} x_{ij}^{k} c_{ij} \tag{7}$$

Subject to :

$$\sum_{k\in K} {y_i}^k = 1, \qquad \forall (i \neq 1) \in N \tag{8}$$

$$\sum_{i>1} {y_i}^k \leq |j| \times {y_0}^k, \qquad \forall k \in K \tag{9}$$

$$\sum_{i\in N} x_{ij}^{k} = {y_j}^k, \qquad \forall j \in N, \forall k \in K \tag{10}$$

$$\sum_{j\in N} x_{ij}^{k} = {y_i}^k, \qquad \forall i \in N, \forall k \in K \tag{11}$$

$$\sum_{ij\in S} x_{ij}^{k} \leq |S| - 1, \qquad (k \in K, \quad S \subset N \quad 2 \leq |S| \leq n-2). \tag{12}$$

Each customer must be visited only once, which is ensured by the constraint (8). The constraint (9) ensures that each round passes through the deposit. The constraints (10) and (11) ensure that each client arrives and leaves. Finally, we find the constraints of elimination of the sub-turns in (12).

2.5.2.2 Variants of the Vehicle Routing Problem

There are different classes or variations of VRP like [95], [96]:

- **Capacitated VRP (CVRP):** a fleet of identical vehicles located at a central depot has to be optimally routed to supply a set of customers with known demands. CVRP is similar to VRP, but in CVRP each vehicle has a capacity Q and a weight, q_i , is associated to each request. The mathematical formulation of this problem is identical to the previous one ((7) to (12)) with the addition of constraint (13) that ensures the fulfillment of the capacity constraint.

$$\sum_{i \in N} q_i {y_i}^k \leq Q, \qquad \forall k \in K \tag{13}$$

- **VRP with Time Windows (VRPTW):** This is the VRP with time windows. It specifies that each client has a time window, which is a period of time during which its service (e.g. loading or unloading goods, visiting a target, etc.) must be completed. Time windows can be of two types: hard and soft. Hard windows represent the constraints of the problem and must absolutely be respected, whereas failure to respect a flexible window leads to a cost penalty, as represented by (14) and (15).

$$a_i \leq u_i^k \leq b_i \qquad \forall i \in N \quad and \quad \forall k \in K \tag{14}$$

$$u_i^k + S_i + t_{ij} - M \times (1 - x_{ij}^k \leq u_j^k) \qquad \forall i \in N, \quad \forall (j \neq 1) \in N, \quad \forall k \in K. \tag{15}$$

The constraint (14) ensures that each request is served in its time window. The stress (15) ensures the time consistency of the rounds. Indeed, it verifies that for two tasks executed consecutively by the same resource, i) the beginning of the second task begins after the end of execution of the first task, ii)the travel time necessary to reach the second site.

2.5.2.3 Application of the VRP for the planning of UAV paths

Mission planning for UAVs is one of the areas where the VRP sees its real application for UAV path planning. In this context, the VRPTW was adapted in [97] in a distributed system of UAVs applied for filming a football match. A variant of the vehicles routing problems with time windows is presented in [98] to consider the routing of vehicles with a heterogeneous fleet.

The problem of planning a mission to monitor a terrain by a fixed number of UAVs was modeled by the VRP and its variants. Its objective is to propose a solution that minimizes the cost and duration of the mission while keeping the level of risk below a certain threshold and respecting the constraints of the vehicles.

The breakdown of the problem is divided into three parts:

- Territory modeling (path planning).
- Assigning Targets to UAVs.
- Calculation of the trajectory.

The major problem with modeling using the TSP is that the latter deals with a single UAV, which explains the use of the VRP.

2.5.3 Constraint Satisfaction Problem (CSP)

The problem of constraint satisfaction, or CSP, is a problem in which we are looking for states or objects satisfying a certain number of constraints or criteria. CSPs are the subject of intensive research in both artificial intelligence and operational research [99]. A modeling for the problem of planning of mission of UAVs based on the constraints satisfaction problem with temporal constraints (TCSP) was proposed in [87]. The approach consists in representing the mission in a set of tasks to be carried out by the UAVs. The problem entries can be grouped into two classes:

2.5.3.1 Constraints related to UAVs and the environment

It is the set of constraints relating to the environment where the mission is carried out. They are represented by the following points:

- Time constraints that ensure that a UAV is at a given time executing a single task.
- The logical constraints, e.g., the minimum or maximum altitude to be reached, restrictions on the areas to be visited (obstacles).
- The resource constraints of the sensors, the energy consumption of the UAVs, etc.

2.5.3.2 Mission-Related Constraints

There are many constraints to the mission. The decomposition of the mission into tasks (targets) and the respect of the temporal constraints of the latter are the most common problems for a UAV mission. The objectives taken into account in [87] are: The total flight time of the UAVs, their energy consumption and the total number of UAVs required for the performance of a mission.

2.6 Conclusion

In this chapter, we first presented the vocabulary related to the missions of drones and their planning by mentioning the various constraints to be taken into account by the system of control and planning. In a second step, we have listed the path planning methods discussed in the literature dealing with the case of one or more vehicles. We have also listed some methods for solving this NP-Complex problem.

We realize our study that mission planning for UAVs is a complex issue. It is broken down into several sub-problems, such as path planning, mission delays, etc. Several possible modeling and variants depending on the context of the mission and the environment in which it is carried out are developed to solve this problem and take full advantage of the UAVs. The UAVs' mission planning problems involve several constraints and often concurrent objectives. These objectives result from the nature of the vehicles themselves since they impose several constraints to be taken into account, in addition to the requirements of the planned missions (execution time).

Part Two: Proposed Solution

Chapter 3

Simulator Design

3.1 Introduction

After giving a general overview on the three concepts: Internet of Things, Unmanned Aerial Vehicle, simulation, and defining the mission of UAVs and its characteristics, we will devote this chapter to the present the design of the simulator. We first present the objective of the simulation, then the global architecture of our proposed system. Finally, the different modules of the simulator will be introduced.

3.2 Identification of needs

In order to identify the users' needs, it is necessary first to frame the context of use of the system, to clarify the hypotheses, to fix the problems to be solved and the requirements induced in terms of functionalities.

3.2.1 Identification of the actors of the system

The system will be handled by two main actors; the first will be the administrator who will have the mission of managing the system and its administration, as well as the users of the system who will benefit from the different functionalities of the system.

3.2.2 Functional requirements

The system must allow the user to:

- Introduce a new mission.
- Delineate the area of a mission.
- Add targets to visit.
- Specify the UAV information.
- Display information about each UAV during the mission.
- Show information on the status of a mission.
- Initiate a mission, i.e. UAV mobility.

- Collect environmental data chosen by the user.
- Visualization of the collected data.

3.2.3 Technical requirements

The system must be:

- Be robust and responsive (Performance in terms of response time).
- Have a user-friendly, simple and easy-to-use interface (HMI).
- Plot the collected data (Histogram, Pie, and Line).
- Allow users to download the data collected in EXCEL file format.
- Be expandable.

3.3 Global architecture of the simulator

In this section, we describe the system of an architectural view. The latter comprises three main modules dealing with the:

1. Trajectory planning (the drone route for visiting targets).
2. Collection of environmental data desired by the user.
3. Communication of drones with the base station.

We first present a general description of the components of the system. We will then detail each module separately. Figure 12 below describes the modular architecture of the various parts of our system.

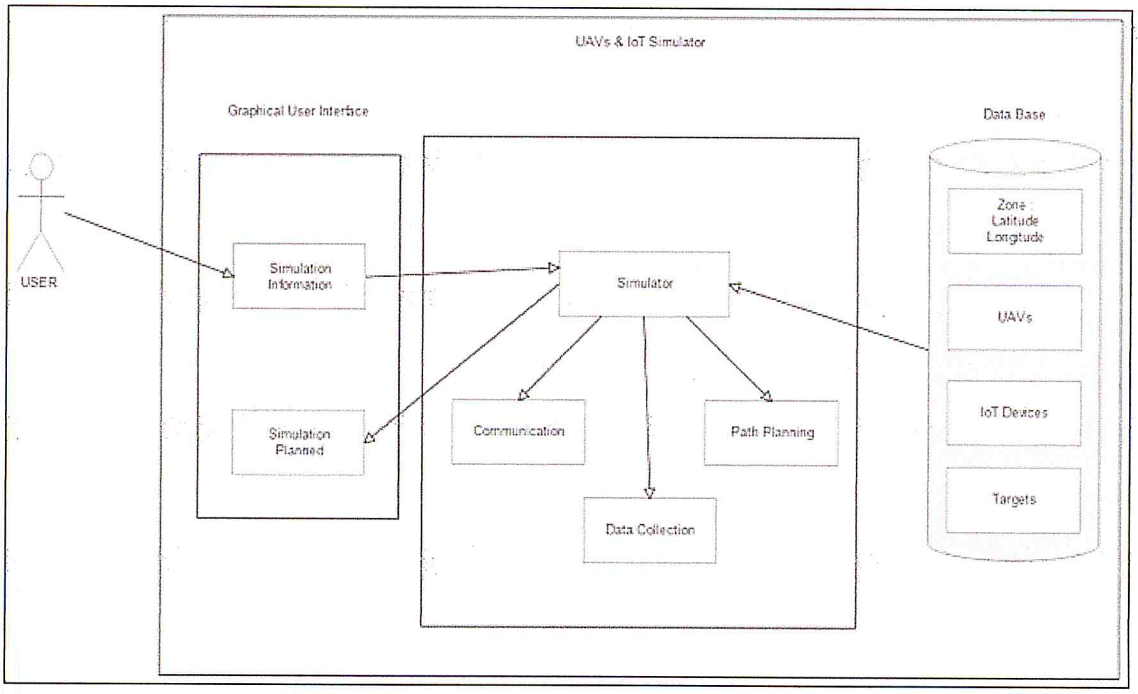


Figure 12: Global architecture of the simulator

The system architecture is made up of three layers; the first layer is the graphical interface that the user uses to interact with the system. Through this interface, the user introduces the missions he wants to accomplish, visualize the course of his mission and the different results obtained.

The second layer represents the system, the part responsible for managing the whole system, namely the planning of the drones trajectories, the data collection, communication, as well as the selection of UAVs.

The last layer is the resource layer. The system resources represent the data that must persist in the system, and which we call upon to realize the different functionalities. This part, therefore, regroups the entities of drones, IoT sensors, the field of the mission and the targets to visit.

3.3.1 Inputs

The following data inputs are used to define a particular scenario and required by the execution of a mission:

- Selection of the mission area.
- Selection of the assigned drone.
- Selection of the average speed.
- Selection of the UAVs ranges which is the area of data collection.
- Selection of the data collection period.
- Introduction of the name of the mission that must be unique.
- Introduction of the number of points (targets) to generate a path.
- Introduction of the time in milliseconds needed to start the mission.
- Selection of the types of data to be collected.

Successfully accomplished tasks have effects assigned to them based on their significance in achieving the objectives outlined for the specific tasks. The effects are specified based on target and data type results.

3.3.2 Outputs

The outputs of the simulation are:

- Scheduling UAVs to the targets (points), i.e. the routes that the UAVs take through their assigned tasks.
- The results of the collected data: temperature, humidity, pressure atmospheric and images.

- The total time of the mission (autonomy of UAVs) which is equal to the sum of sensing and processing, traveling and communication time.
- The total energy which is equal to the sum of sensing and processing, traveling and communication energy.

The mission time (autonomy of the drone) and energy will be calculated based on a mathematical model.

3.4 Modular Design

In this section, we describe the architecture of the system from a modular point of view, so we will first identify the main modules constituting the system, the function of each module and the interactions between them.

3.4.1 Path Planning

The purpose of this module is to define the trajectory of the vehicles for the visit of the targets. The module will take the input targets into account with vehicle information. It will determine two scenarios:

1. The random path of the drones to cover the entire area.
2. Follow a path desired by the user as a second scenario of the path planning.

In this context, we consider that the path between one point and another is a line. The equation of this line is calculated according to the coordinates of the target source and destination. Let A(x1, y1) be the source target and B(x2, y2) the destination, the straight line connecting the two points is a set of points represented by an affine function f(x), such as:
f(x) = ax + b where, b=$(y2 - y1)/(x2 - x1)$

3.4.1.1 Random way point mobility model

The principle of this model is the generation of random paths in the selected area. Thus, the algorithm takes the number of points (at the choice of the user) and generates them randomly in the selected zone. A path is then established, by connecting between each two points.

3.4.2 Data collection

One of the main features of the system is the collection of environmental data. Using the sensors embedded on the UAVs, the latter go in their route to collect the data and transmit it to the base station which will entrust their processing. The collection module takes as input the type of the data to be collected, in our case we limit it to: temperature, humidity, atmospheric pressure and photos as well as, the time of collection and the sampling period (which is converted into intervals separating two successive collections), and generates digital values or satellite photos as outputs.

Once the data has been collected, it will be saved and displayed to the user as graphs (Histogram, Pie, and Line), or an Excel file that he can download for next use.

The format of the data is shown in the following table:

Type	Collection Time	Latitude	Longitude	Values
The type of data collected: - Temperature - Humidity -Atmospheric Pressure - Photos	The moment corresponding to the time data collection	Latitude of UAVs	Longitude of UAVs	Numerical collected of value of the physical quantity with the exception the photo

Table III: Format of collected data

In order to simulate the values of the collected data, we randomly vary the numerical values in an interval that we have even delimited.

As for the collection of photos, we retrieve the actual satellite photos provided from the Google Map API. The Photo Collection module takes as input a URL, which contains the following specifications:

- Google maps API.
- Extent which is a decimal degree: mapping unit.
- Adding and deleting extent from the coordinates.
- The width.
- The height.
- The zoom of the picture.

3.4.3 Communication

Two types of communication exist in a network of drones, i) communication between UAVs and, ii) communication between UAVs and the base station. For our case, we only consider the communication between the base station and the drones.

This section models the communications between a UAV u and a base station (also called eNodeB). In order to improve the communication reliability, an automatic repeat request (ARQ) scheme is used for forwarding the information.

Let u denotes the transmitting UAV, whereas B refer to the receiving eNodeB. The channel gain between u and $\mathcal{B}$ is referred to as $\alpha_{u,\mathcal{B}}$. A Rayleigh block-fading channel is considered, where the channel gain $\alpha_{u,\mathcal{B}}$ remains constant over one block but changes independently from one block[1] to another. The received signal $y_{\mathcal{B}}$ at a destination node B can be expressed as [63]:

1. A block corresponds to the time duration necessary to send one packet.

$$y_{\mathcal{B}} = \alpha_{u,\mathcal{B}}\sqrt{P_u}x_u + n_{\mathcal{B}}, \tag{16}$$

where P_u denotes transmission powers of UAV u. The symbols transmitted by node u is referred to as x_u. $n_{\mathcal{B}}$ is a zero-mean additive white Gaussian noise with variance N_0. The term $\gamma_{u,\mathcal{B}}$ denotes the instantaneous received signal-to-noise ratio at $\mathcal{B}$ given by $\gamma_{u,\mathcal{B}} = P_u\alpha^2_{u,\mathcal{B}}/N_0$ [100]. The mean value of $\gamma_{u,\mathcal{B}}$ is denoted as $\bar{\gamma}_{u,\mathcal{B}}$ which can be expressed as:

$$y_{u,\mathcal{B}} = \frac{P_u E[\alpha^2_{u,\mathcal{B}}]}{N_0}, \tag{17}$$

Where $E[\alpha^2_{u,\mathcal{B}}]$ represents the channel variance, and E[.] is the expectation operator. Using a distance dependent path loss model, the channel variance can be determined as [101]:

$$E[\alpha^2_{x,y}] = \left(\frac{\mathcal{D}_0}{\mathcal{D}_{u,\mathcal{B}}}\right)^{\beta}, \tag{18}$$

Where $D_{u,B}$ refers to the distance between nodes u and B, D_0 denotes a reference distance typically set to 1 m, and β denotes the path loss exponent. In our physical model, we take into account the effects of both path loss and fast fading, while the impact of shadowing is neglected.

Theorem 1
For any UAV $u \in \mathcal{N}$fails to transmit its packet to an eNode$\mathcal{B}$ iff $\mathsf{SNR}_{u,\mathcal{B}}$ falls below a threshold γ_{th}. This event is known as an outage event and occurs with a probability $\mathcal{P}_{u,\mathcal{B}}$ which can be expressed as [63]:

$$\mathcal{P}_{u,\mathcal{B}} = 1 - \exp\left(-\frac{\gamma_{\text{th}}}{\bar{\gamma}_{u,\mathcal{B}}}\right). \tag{19}$$

3.5 Conclusion

In this chapter, we explored the functional aspects of our solution in more detail. After describing the overall modular architecture of our system, we explained the phase of needs analysis in terms of functional, technical and operational scenario specifications. We then detailed the design of each module of the system in order to achieve the set objectives.

In the next chapter, we will describe the realization and implementation of our system, as well as the technical choices we have made.

Chapter 4

Realization and Test

4.1 Introduction

After presenting the design of the proposed system and its different modules, we present its implementation in this Chapter.

First, we present the different technical choices (operating system, development environment and programming language) that we faced in the implementation. Finally, we will detail the phase of tests conducted in order to concretize our design and to identify the most suitable methods .

4.2 Development Environment

We are deploying the system designed for a Linux operating system with the UBUNTU distribution (16.04 LTS) using the Python and JavaScript as a programming language. We have also used open source tools and libraries that will be detailed in the following.

4.3 Tools

4.3.1 Editors

We used the following editors:

a. PyCharm IDE:
PyCharm is an Integrated Development Environment (IDE) used in computer programming, specifically for the Python language. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django. It is cross-platform, with Windows, Mac OS and Linux versions [102].

b. Sublime text:
Sublime Text is a proprietary cross-platform source code editor with a Python application programming interface (API). It natively supports many programming and markup languages. Functions can be added by users with plugins, which are typically community-built and maintained under free-software licenses [103].

4.4 Programming Languages

4.4.1 Python

Python is an interpreter, interactive, object-oriented programming language. It provides high-level data structures such as list and associative arrays (called dictionaries), dynamic typing and dynamic binding, modules, classes, exceptions, automatic memory management, etc. It has a remarkably simple and elegant syntax and yet is a powerful and general purpose programming language [104].

4.4.2 JavaScript

JavaScript is an interpreted programming language with Object-Oriented (OO) capabilities [105]. It is used to make web pages interactive and provide on-line programs, including video games. The majority of websites embed JavaScript codes, and all modern web browsers support it without the need for plug-ins by means of a built-in JavaScript engine.

We used also:

- jQuery which is a JavaScript library [106] that allows the user to program many different cool effects on the web pages, including color changes, animations, and fade in/fade out effects.
- TypeScript is an extension of JavaScript intended to enable easier development of large-scale JavaScript applications [107]. It is pure object oriented with classes, interfaces and statically typed, similarly to C# or Java.

4.4.3 Framework and Libraries

4.4.3.1 Django

Django is a high-level Python Web framework that facilitates rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development. We can focus on writing the application without needing to reinvent the wheel [108]. It's free and open source.

Django is inspired by the MVT (Model View Template) model, where the structure of the framework separates the data (models) from the processing (controller), which is separated from the view/template [109].

Django's choice was motivated by the following reasons:

- Simplicity of learning.
- The quality of the applications carried out.
- Speed of development.
- The security of the final website.
- Ease of application maintenance over time.

4.4.3.2 Google Maps

Google Maps is a Google service offering powerful, user-friendly mapping technology and local business information, including business locations, contact information, and driving directions [110]. It offers satellite imagery, street maps, 360 deg panoramic views of streets (Street View), real-time traffic conditions (Google Traffic), and route planning for traveling on foot, by car, bicycle (in beta), or public transportation.

The Google Maps service's interface utilizes JavaScript, XML, and Ajax. Google Maps offers an API that allows maps to be embedded on third-party websites and offers a locator's for urban businesses and other organizations in numerous countries around the world [111].

4.4.3.2.1. Google Map JavaScript API

Available for web services via HTTP. The Google Map API is a free service that offers a wide range of applications and utilizations that include localization around the world by simply introducing markers and also allow developers to integrate Google Maps into their websites. Google Maps APIs are also available for Android, iOS, Web browsers [112].

4.5 RESTful API

A RESTful API is an Application Programming Interface (API) that uses HTTP requests to GET (retrieve a resource), PUT (change the state of or update a resource, which can be an object, file or block), POST (create that resource) and DELETE (remove resource) data [113]. An API for a website is code that allows two software programs to communicate with each other. It spells out the proper way for a developer how to write a program requesting services from an operating system or other application.

A RESTful API – also referred to as a RESTful web service – is based on REpresentational State Transfer (REST) technology, and client-server architecture style [114]. REST technology is generally preferred to the more robust Simple Object Access Protocol (SOAP) technology as it leverages less bandwidth, making it more suitable for Internet usage.

The REST used by browsers can be thought of as the language of the internet. With cloud use on the rise, APIs are emerging to expose web services. REST is a logical choice for building APIs that allow users to connect and interact with cloud services. RESTful APIs are used by such sites as Amazon, Google, LinkedIn, and Twitter.

4.5.1 How RESTful APIs Work

A RESTful API breaks down a transaction to create a series of small modules. Each module addresses a particular underlying part of the transaction [115]. This modularity provides developers with a lot of flexibility, but it can be challenging for developers to design from the scratch. Currently, the models provided by Amazon Simple Storage Service, Cloud Data Management Interface, and OpenStack Swift are the most popular.

4.6 System Implementation

Our simulator is a web application developed under Django that aims to simulate a mission of the Unmanned Aerial Vehicles (drones) and its behaviors. This mission consists of the visit, by one or several drones, of several targets (points) to carry out a task over a given period. Each point of the zone is defined by its geographical coordinates, latitude and longitude, according to which the data must be collected. For this, we consider Google Maps as an environment in which we can test and evaluate the different results of a mission of these air vehicles.

Energy efficiency and time are the crucial factors for the success and wide acceptance of the drone technology. In this vein, it is important to indicate the time and the energy consumed by UAVs in fulfilling a particular IoT task.

4.6.1 Land modeling

Our Google maps interface which is a mapping service contains two display modes: plan and satellite. Our choice goes on Google maps as it is a free service that covers the whole world and that offers many useful features such as geo-location and resolution of satellite images, it is well documented, etc.

To use this service, we first set up all Google Maps JavaScript API configurations, starting with creating a project and having the key, then activating the APIs and adding the necessary libraries for the development.

The HTML code below represent the creation of base Google Maps.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Simple Map</title>
    <meta name="viewport" content="initial-scale=1.0">
    <meta charset="utf-8">
    <style>
      /* Always set the map height explicitly to define the size of the div
       * element that contains the map. */
      #map {
        height: 100%;
      }
      /* Optional: Makes the sample page fill the window. */
      html, body {
        height: 100%;
        margin: 0;
        padding: 0;
      }
    </style>
  </head>
  <body>
    <div id="map"></div>
    <script>
      var map;
      function initMap() {
        map = new google.maps.Map(document.getElementById('map'), {
          center: {lat: -34.397, lng: 150.644},
          zoom: 8
        });
      }
    </script>
    <script src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap"
    async defer></script>
  </body>
</html>
```

4.6.2 Database setup

Django in its 'out-of-the-box' state is set up to communicate with SQLite – a lightweight relational database included with the Python distribution. So by default, Django automatically creates a SQLite database for the project.

The figure 13 illustrates django site administration.

جامعة البليدة 1 - كلية العلوم
مكتبة
الكلية

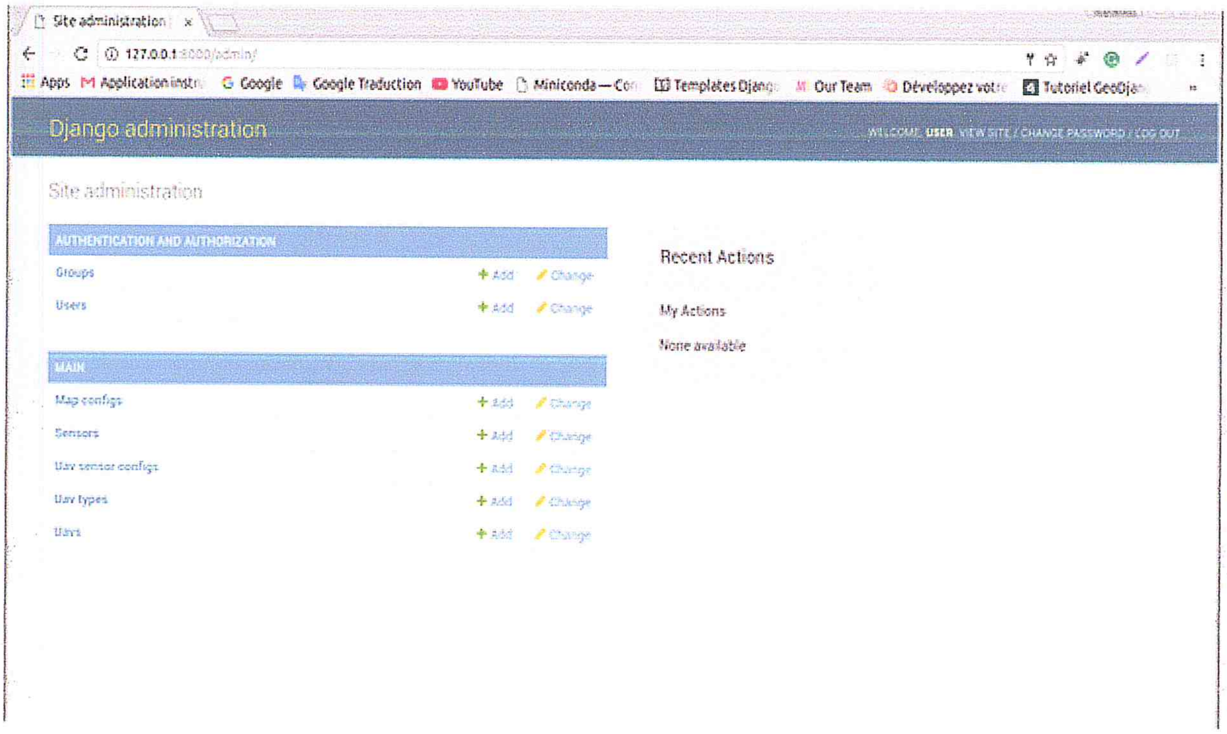


Figure 13: Django site administration

In addition to SQLite, Django also has support for other databases, e.g., PostgreSQL, MySQL and Oracle. The Django configuration to connect to a database is done inside the setting.py file of a Django project in the DATABASES variable. DB Browser for SQLite is a high quality, visual, open source tool to create, design, and edit database files compatible with SQLite.

4.6.3 UAV modeling

The main entity of the system is that representing the UAVs. The main attributes of the UAV class are listed in the following:

- ID of UAV.
- UAV label.
- CPU (GHZ).
- Range of UAVs (m).
- Weight (kg).
- Autonomy (minute) or lifetime, which is the period that the drone can fly.
- Connexion.
- Storage Capacity (GO).
- Battery of UAV (%).

- Speed (m/s).
- Type of UAVs (Micro, mini, tactical and strategic).

4.6.4 Simulator modules

For UAV selection mechanism, let u denote to a UAV, $\mathcal{N}$ be a set of UAVs in the network, $\mathcal{E}$ denote to an event, and $u \in \mathcal{N}$.

4.6.4.1 Path Planning

Path planning provides a set of way points defining the routes that drones can take to reach targets. During the mobility, each drone follows a path from the point of departure to the point of arrival. To do this, it will move either to the left or to the right, following the drawn path or a path established randomly. For the generation of random paths, we wrote an algorithm, which takes a number of points (desired by the user) as an input, and at the execution of this code, it randomly puts the points, that they must be in the selected zone, then to have a path, we must connect between each two points to have a straight line.

The python code below represent the random point generation

```
def get_point(lat1, lng1, lat2, lng2, paths, n, i=1):
    if i <= n:
        lat = random.uniform(lat1, lat2)
        lng = random.uniform(lng1, lng2)
        point = {
            'lat': lat,
            'lng': lng
        }
        paths.append(point)
        get_point(lat1, lng1, lat2, lng2, paths, n, (i+1))


def get_paths(lat1, lng1, lat2, lng2, num_of_points):
    paths = []

    if num_of_points < 2:
        num_of_points = 2

    get_point(lat1, lng1, lat2, lng2, paths, int(num_of_points))
    return paths
```

The generation of the points is on the server part (back-end), and to recover the results and to display them in the front-end, we proceed with the architecture RESTful API.

The JavaScript code below represent the generation of point from the server.

```
...
/**
 * Create a new mission
 * @type {Mission}
 */
var startTime = Utils.simulationConfigForm.find("#mission-start-time").val();
var mission = new Mission(missionName, uav, zone, startTime);
Utils.missions.push(mission);
/**
 * Get points from the server
 * @type {string}
 */
var numPointsToGenerate = Math.floor(+Utils.simulationConfigForm.find("#mission-number-of-points-to-generate").val());
var url = "http://127.0.0.1:8000/api/paths/" + mission.zone.P1.Lat + "/" + mission.zone.P1.Lng + "/"
+ mission.zone.P2.Lat + "/" + mission.zone.P2.Lng + "/" + numPointsToGenerate + "/" + mission.zone.getXLength() + "/"
+ mission.zone.getYLength();
if (numPointsToGenerate < 2) {
    toastr.error("The number of points needs to be greater or equal to 2 points");
    return false;
}
$.ajax(url, {
    success: function (response) {
        var coords = [];
        var red = Math.floor((Math.random() * 255));
        var green = Math.floor((Math.random() * 255));
        var blue = Math.floor((Math.random() * 255));
        var strokeColor = "rgba(" + red + ", " + green + ", " + blue + ", 1)";
        response.points.forEach(function (point) {
            var latLng = new LatLng(point.lat, point.lng);
            coords.push(latLng);
        });
```

4.6.4.1.1 Energy consumption and required time for traveling

The largest amount of the energy consumed by the UAV is the amount of the energy that a UAV utilizes to travel to an event position. This energy is called the Traveling Energy of the UAV, ξ_u^{Travel}.

In order to compute the travel energy, the travel distance $\mathcal{D}(u, \mathcal{E})$ between UAV and the event position should be calculated. Using the average velocity $\mathcal{V}_u$, the traveling time Υ_u^{Travel} can be computed.

For calculating the travel energy ξ_u^{Travel}, the coefficient λ, which represents the amount of energy consumption per one meter should be computed. It can be considered as the proportion of the full battery amount $\xi_u^{Battery}$ to $\mathcal{D}_u^{Range}$ which is the maximum distance that a UAV can fly.

Considering $\mathcal{P}_{\mathcal{E}} = (X_{\mathcal{E}}, Y_{\mathcal{E}})$ and $\mathcal{P}_u = (X_u, Y_u)$, positions of an event and the position of a UAV in 2-D space respectively, and having $\Upsilon_u^{Endurance}$, the UAV endurance time. The traveling time and energy of a UAV can be formulated as [63]:

$$\mathcal{D}(u, \mathcal{E}) = \sqrt{(X_u - X_{\mathcal{E}})^2 + (Y_u - Y_{\mathcal{E}})^2} \tag{20}$$

$$\Upsilon_u^{Travel} = \frac{\mathcal{D}(u, \mathcal{E})}{\mathcal{V}_u} \tag{21}$$

$$\mathcal{D}_u^{Range} = \mathcal{V}_u \times \Upsilon_u^{Endurance} \tag{22}$$

$$\lambda = \frac{\xi_u^{Battery}}{\mathcal{D}_u^{Range}} \tag{23}$$

$$\xi_u^{Travel} = \lambda \times \mathcal{D}(u, \mathcal{E}) \tag{24}$$

4.6.4.2 Data Collection

During the mobility of the drones, the desired data (temperature, atmospheric pressure, humidity) is collected (sensed and processed) by the user (of the choice). These data are generated randomly in a range of values limited by the user in its implementation. For photos, the simulator generates satellite photos.

The method of collection and some collection functions are presented in the following:

```
class Collector:
    temperature_val = {'min': 20, 'max': 30}
    humidity_val = {'min': 50, 'max': 60}
    atmospheric_pressure_val = {'min': 800, 'max': 1000}

    photo_val = {
        'width': 400,
        'height': 400,
        'delta': 0.1,
        'zoom': 18
    }

    def __init__(self, what, time, lat=0, lng=0):
        self.what = what
        self.time = time
        self.lat = lat
        self.lng = lng

    def temperature(self):
        return {
            'Type': 'Temperature',
            'Time': self.time,
            'Value': random.uniform(self.temperature_val.get('min'), self.temperature_val.get('max')),
            'Lat': self.lat,
            'Lng': self.lng
        }
```

About the image, it's a real photo recover from Google maps satellite using an API. The python code for image collection is represented below:

```
def photo(self):
    url = "http://maps.googleapis.com/maps/api/staticmap?sensor=false&size=" +\
          str(self.photo_val.get('width')) +\
          "x" +\
          str(self.photo_val.get('height')) +\
          "&maptype=satellite&visible=" +\
          str((self.lat - self.photo_val.get('delta'))) +\
          "," + str((self.lng - self.photo_val.get('delta'))) +\
          "&visible=" +\
          str((self.lat + self.photo_val.get('delta'))) +\
          "," + str((self.lng + self.photo_val.get('delta'))) +\
          "&zoom=" + str(self.photo_val.get('zoom'))


    return {
        'Type': 'Photo',
        'Time': self.time,
        'Value': url,
        'Lat': self.lat,
        'Lng': self.lng
    }
```

6.4.2.1. Energy consumption and required time for sensing and processing data

Any sensor which is applied with UAVs needs its own energy consumption amount and in some cases its own sampling frequency. Generally, it is assumed that a sensor has the following sensing energy consumption:

$$\xi_u^{Sensing} = V_{dc} \times I_i \times T_i \tag{25}$$

where T_i is the time needed for obtaining a single sample from the sensor i and I_i is the current draw of sensor. In order to compute the processing energy of the sensors $\xi_u^{Processing}$, generally MCU's (Micro Controller Unit) active and sleep modes for currents and times are used as in the following equation [116]:

$$\begin{aligned}\xi_u^{Processing} = V_{dc} \times I_{mcu-active} \times T_{mcu-active} + \\ V_{dc} \times I_{mcu-sleep} \times T_{mcu-sleep}\end{aligned} \tag{26}$$

Then the energy and time of sense-and-process will be:

$$\xi_u^{SensProcess} = \xi_u^{Sensing} + \xi_u^{Processing} \tag{27}$$

$$\Upsilon_u^{SenseProcess} = T_i + T_{mcu-active} \tag{28}$$

4.6.4.3 Communication

The drones are constructing a Flying Ad-hoc NETwork (FANET) in which they communicate in an ad-hoc manner to obtain an efficient UAV networking. They also can communicate with the base station via a cellular. Popular cellular System includes Global System for Mobile communication (GSM), universal mobile telecommunication system or 3G, and LTE or 4G. Our implementation is based on 4G LTE technology, to allow the drones to communicate with the base station or what is called eNodeB.

4.6.4.3.1. Communication time modeling

The UAVs are equipped with a buffer to store the packets before their transmission. We assume that UAV has $\mathcal{K}$ packets that represent the sensed information about an event $\mathcal{E}$. This section focuses on the analysis of the average delay of transmission Υ_u^{Travel} of sensed data from to an eNodeB [1] $\mathcal{B}$. . We do not take into account neither the time required for sensing and processing nor the time required for traveling from the even $\mathcal{E}$ to $\mathcal{B}$.

Let $T_u^{Transmit}$ represents the sojourn time of a packet before its transmission to $\mathcal{B}$. Formally: $\Upsilon_u^{Transmit} = \mathcal{K} \cdot T_u^{Transmit}$.

A successful reception of a packet at eNodeB $\mathcal{B}$ occurs after a random number of retransmissions. To quantify the delay associated with the retransmission events, we measure the average sojourn time $T_u^{Transmit}$ of a packet in the buffer of u ,which is defined as the average time elapsed from the starting of its transmission until its successful reception. The packet's sojourn time in the buffer can be evaluated using the Pollaczek-Khinchin equation [117]:

1. The base station for mobile networks based on LTE or LTE Advanced technologies.

$$T_u^{Transmit} = E(N_{u,\mathcal{B}})T_F \tag{29}$$

where T_F is the time required for a single transmission of a given packet and $E(N_{u,\mathcal{B}})$ is the average number of retransmissions for the packets sent by u .For the ARQ scheme, the packet is retransmitted until successful reception at the $\mathcal{B}$ or a maximum number of retransmissions M is reached.

In case of reception failure after M retransmissions the packet is discarded. The number of retransmissions $N_{u,\mathcal{B}}$ varies randomly according to the position of UAV and the conditions of the fading channel between the u and $\mathcal{B}$. The average number of retransmissions $E(N_{u,\mathcal{B}})$ can be expressed as [118]:

$$E(N_{u,\mathcal{B}}) \;=\; 1+\sum_{m=1}^{M-1} P(F^1,...,F^m) = 1+\sum_{m=1}^{M-1}(\mathcal{P}_{u,\mathcal{B}})^m = \sum_{m=0}^{M-1}(\mathcal{P}_{u,\mathcal{B}})^m = \frac{1-(\mathcal{P}_{u,\mathcal{B}})^M}{1-\mathcal{P}_{u,\mathcal{B}}}, \tag{30}$$

where $P(F^1,...,F^m)$ is the probability of a reception failure at the $1,\ldots,m^{th}$ retransmissions. Since the channel realizations in each transmission are independent identically distributed random variables, the event of reception failure at each step are independent and have equal probabilities, thus $P(F^1,...,F^m) = (\mathcal{P}_{u,\mathcal{B}})^m$. From 29 and 30, we have:

$$y_u^{Transmit} = \mathcal{K}\cdot T_F \cdot E(N_{u,\mathcal{B}})\cdot \frac{1-(\mathcal{P}_{u,\mathcal{B}})^M}{1-\mathcal{P}_{u,\mathcal{B}}}. \tag{31}$$

4.6.4.3.2 Energy consumption modeling in communication

This section studies the energy consumption at UAV u. Let's assume that a UAV has K packets that represent the sensed information about an event $\mathcal{E}$. The fact that the number of retransmissions varies depending on the channel conditions makes the consumed power a random variable. This section studies the average consumed power and then deduce from that the average consumed energy. The average consumed power $\bar{P}$ for the ARQ scheme can be determined as:

$$\begin{aligned}\bar{P} &= P_u.P(S^1) + 2P_u.P(F^1,S^2) + \cdots + (M-1)P_u.P(F^1,...,S^{M-1}) + MP_x..P(F^1,...,F^{M-1})\\ &= P_u.\left(1+\sum_{m=1}^{M-1} P(F^1,...,F^m)\right) = P_u.\left(1+\sum_{m=1}^{M-1}(p_{u,\mathcal{B}})^m\right) = P_u.\mathbb{E}(T_{u,\mathcal{B}})\\ &= P_u.\frac{1-(p_{u,\mathcal{B}})^M}{1-p_{u,\mathcal{B}}}\end{aligned} \tag{32}$$

where the term P_u stands for the power per retransmission at a given u. We denote by $P(S^1)$ the probability of successful reception at $\mathcal{B}$ of the first transmission, while $P(F^1, ..., S^{M-1})$ refers to the probability of a reception failure in the 1^{st}, 2^{nd}, ..., $(M-2)^{th}$ retransmissions and a successful reception at the $(M-1)^{th}$ retransmission.

If the packet is successfully received after the first transmission (this event occurs with a probability $P(S^1)$), the amount of consumed power would be equal to P_u. If the packet is received correctly after two retransmissions (the probability of this event is $P(F^1, S^2)$), the amount of consumed power would be equal to $2P_u$.

The consumed power would be equal to MP_u if the 1^{st}, ..., $(M-1)^{th}$ retransmissions fails (the probability of this event is $P(F^1, ..., F^{M-1})$). The average consumed power is obtained by summing up all the possible values of consumed power weighted by their respective probability of occurrence.

The result in 32 shows that we can express the average consumed power as the product of two terms: the power per retransmission P_u and the average number of retransmission $E(T_{u,\mathcal{B}})$.The average consumed energy $\Phi_{u,\mathcal{B}}$ of one packet can be obtained as:

$$\begin{aligned}\Phi_{u,\mathcal{B}} &= P_u.T_F.P(S^1) + 2P_u.T_F.P(F^1,S^2) + \dots + (M-1)P_u.T_F.P(F^1,\dots,S^{M-1}) \\ &+ MP_u.T_F.P(F^1,\dots,F^{M-1}) = P_u.T_F.\left(1+\sum_{m=1}^{M-1}(p_{u,\mathcal{B}})^m\right) + P_u.T_F.\mathbb{E}(T_{u,\mathcal{B}}) = \bar{P}.T_F\end{aligned} \tag{33}$$

The average consumed energy $\xi_u^{Transmit}$ to transmit the whole data ($\mathcal{K}$ packets) can be obtained as:

$$\xi_u^{Transmit} = \mathcal{K} \cdot \Phi_{u,\mathcal{B}} = \mathcal{K} \cdot \bar{P} \cdot T_F. \tag{34}$$

The total energy and the time of mission are equal to the sum of sensing and processing, traveling and communication energy, as representing below:

```
# The total energy is the sum of the energy required for the collect (EnergySensProcess), travel and of the transmission
# Equation 3
def getEnergyTotal(self):
    return self.getEnergySensProcess() + self.getEnergyTravel() + self.getEnergyTransmit()

# The total time is the sum of .. (as energy)
def getTimeTotal(self):
    return self.getTimeSensProcess() + self.getTimeTravel() + self.getTimeTransmit()
```

4.6.5 Data Export

The export of data is the automated or semi-automated input and output of data sets between different software applications.

Export of data shares a semantic analogy with copying and pasting, i.e., sets of data are copied from one application and pasted into another, by specifying attributes such as field names, field lengths, and field data type, etc.

A sample code for exporting data is shown below:

```
class Exporter:

    temperature = []
    humidity = []
    atmospheric_pressure = []
    photo = []

    def __init__(self, prefix, mission_name, data):
        self.prefix = prefix
        self.mission_name = mission_name
        self.data = data
        self.filename = self.prefix + self.mission_name + ".xlsx"

        if os.path.exists(self.filename):
            os.remove(self.filename)

        self.workbook = xl.Workbook(self.filename)
        pass

    def export(self):
        self.prepare_data()

        self.export_temperature()
        self.export_humidity()
        self.export_atmospheric_pressure()
        self.export_photo()

        self.workbook.close()
```

4.6.6 Exception Management System

Exception Management System or EMS allows to manage the exceptional conditions during the execution of the program. When an exception occurs, normal program execution is aborted and the exception is processed. An exception handler sets up a set of error-defined routines defined by the programmer on a block (in a function or a program method). These routines are activated for the duration of the execution of the protected block. In the following, we present some exceptions of our implementation:

- The name must be entered for each selected zone or the designated path.
- The name of each path and the zone must be unique.
- To display the results of the data collection in graphs, we have to choose the type of the data, as well as the type of the graph (except in the case of photos when the type of graphs is not selected).
- Each mission has a unique name.
- Each drone cannot assign to another mission until the mission is completed.
- Another important exception is related to the energy. After configuring the mission parameters, the program calculates the total energy consumed by the drone (sensing, processing, traveling and communication energy) and subtracts this value from

the initial value of the battery. If a a non-positive value is obtained, the mission does not start (i.e. for the success of a mission, we should first check whether the drone has enough energy to perform a given task).

4.7 Test

In the following, we test the behavior of UAV that performs a mission in the context of a surveillance application and data collection over the zone. Deploying UAVs for this task is favorable due to their long endurance and vast area of coverage. The UAVs mission consists of sending two drones to monitor the zone where each of them is responsible for capturing and taking photos during their mobility, as well as collecting the data (temperature, atmospheric pressure and humidity).

This scenario is of particular interest as it is a potential application for UAVs and well-suited to evaluate the reliability of this drones on reflecting the real environment.

The final outputs of the mission planning systems are flying paths for the UAVs to carry out the surveillance of a zone and recover the collected data.

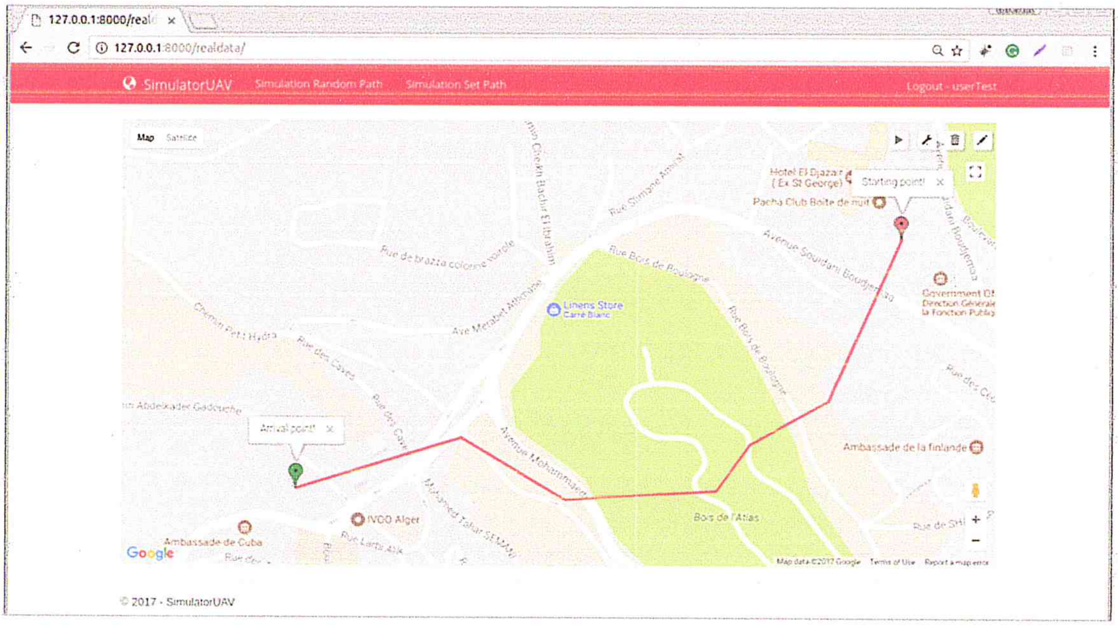
The main features that have been investigated are:

- Path generation:

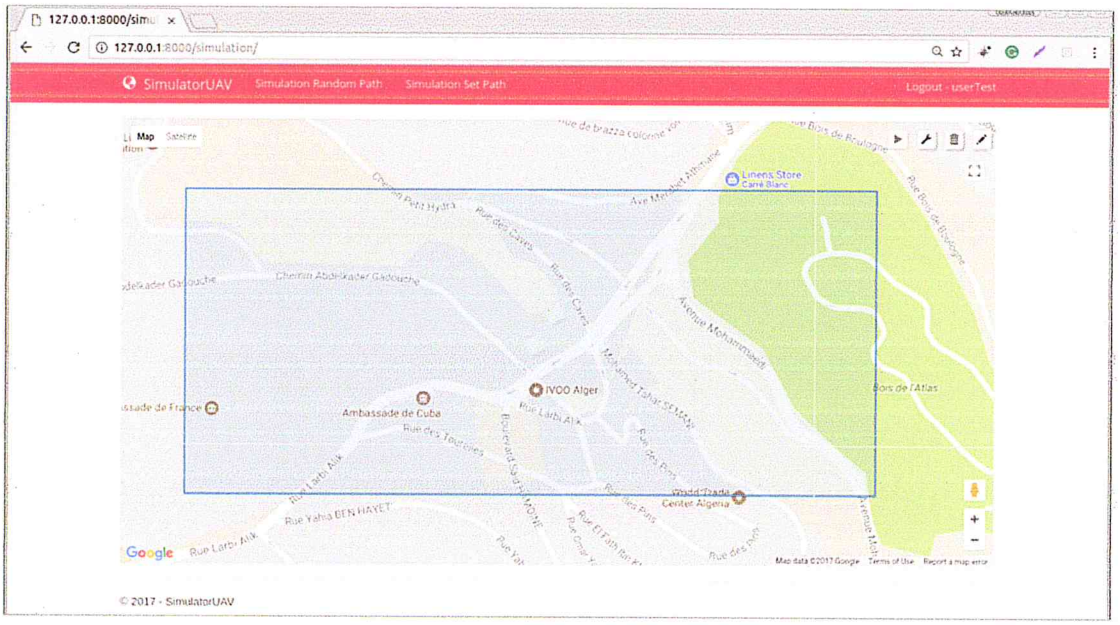
  After the authentication of the user, he chooses the type of the simulation desired, i.e. simulation with a set of paths by drawing with the mouse a path from the departure point to the arrival point as showing in the figure 14 (a) or, the simulation with a random path by selecting the area on the Google maps as demonstrating in the figure 14 (b).

  In the both scenario, every path or zone should have an unique name. The generation number of path and zone is depending on the number of UAVs in the database.

  In the following, we will illustrate an example of the random path simulation.



(a)



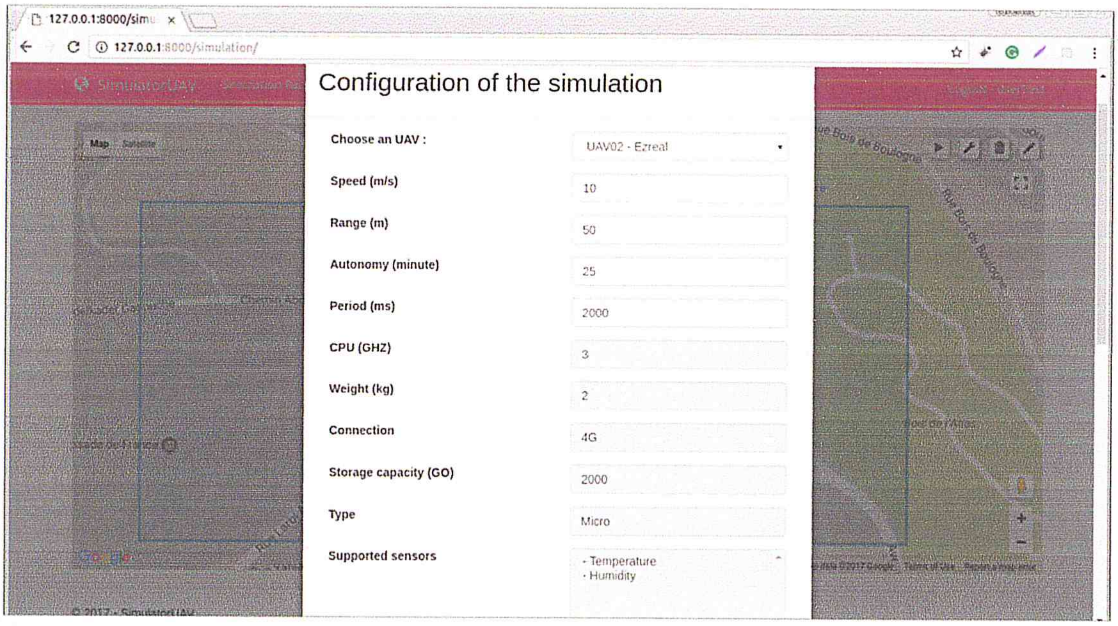
(b)

Figure 14: GUI for path generation

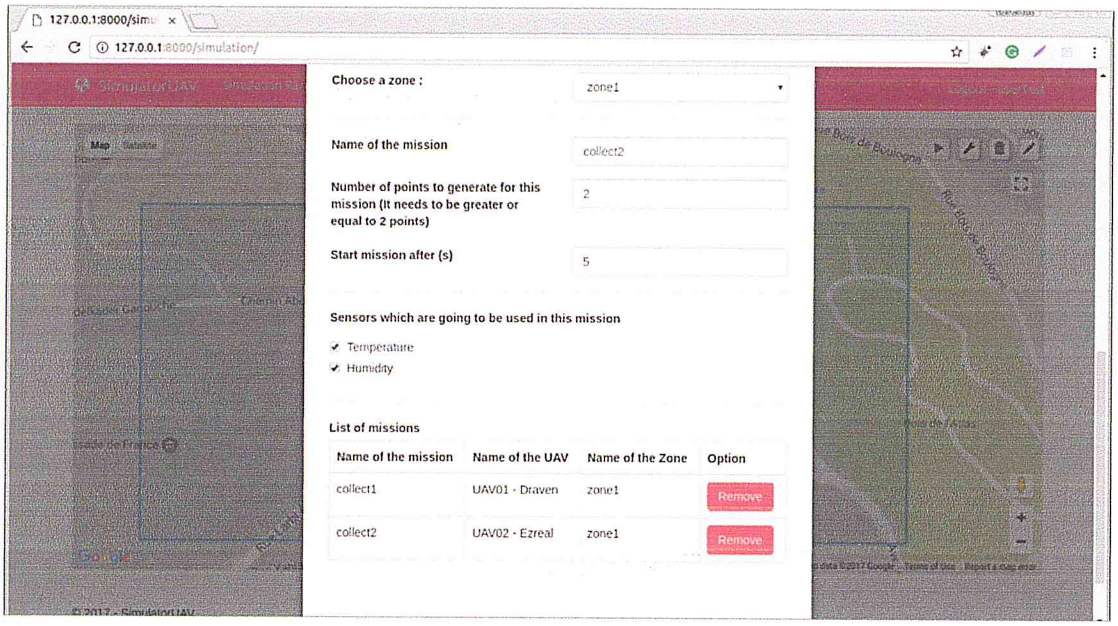
- Configuring the settings of a mission:

  The configuration of a mission is to select the drone from an existing list in the DB, its speed, its range, the mission area (in the case of the selection of several zones), the name of the mission, the type of data collected, the collection period, the number of points to generate the path, and the time required to start the mission.

The following mission shows the sending of two drones in the same zone, and each one has its own speed, range, IoT sensor. The first drone follows a randomly generated path of 8 points and begins 10s after launching a mission (by clicking a start button), and the second drone follows a path with 2 points and begins after 5s of the launch of the mission and sensible to capture the data every 2s. The configuration form for this UAVs mission is presented in the figure 15 (a,b).



(a)



(b)

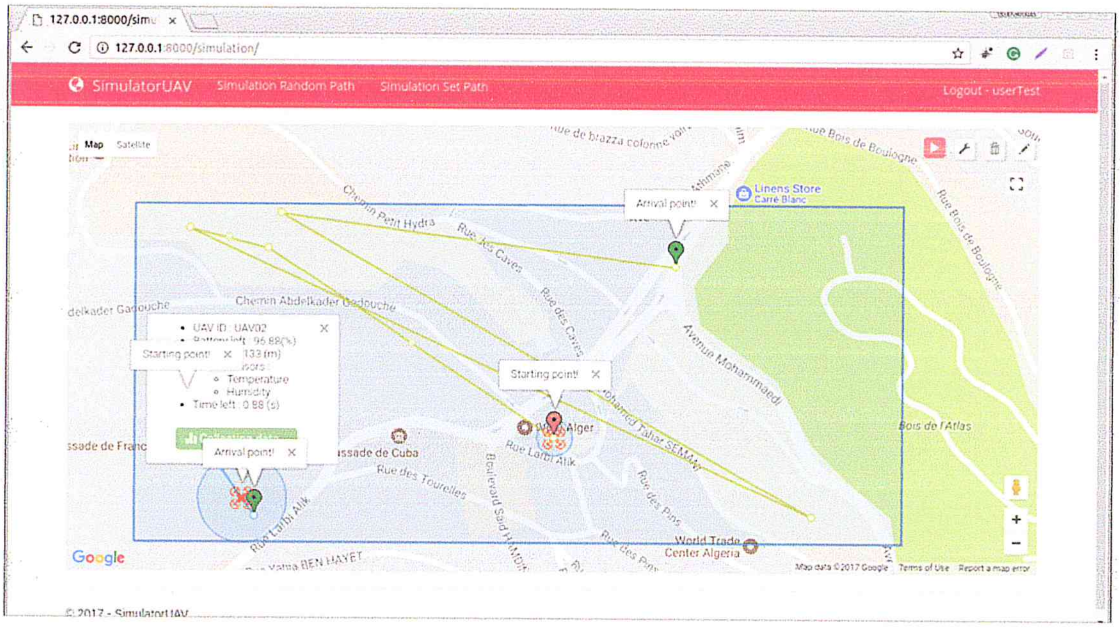
Figure 15: Display Window for the configuration of the UAVs mission

- Mobility and data collection:

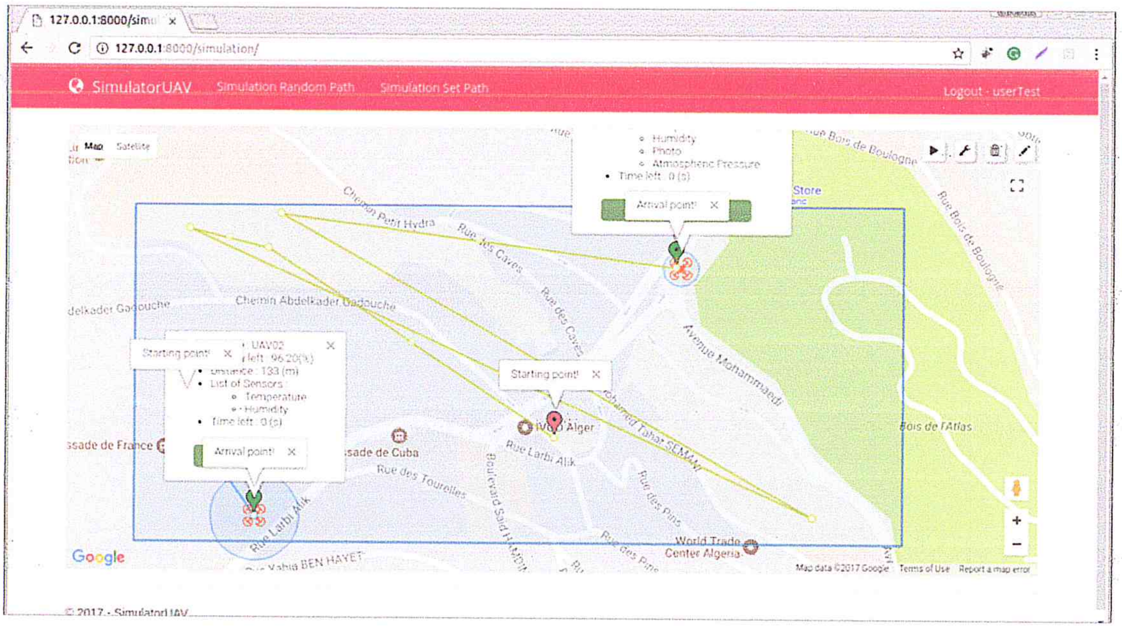
  After saving the simulation parameters, we click on the button to start the simulation. The second drone start moving before the first one because he start the mission after 5s on the other hand the second drone start after 10s.

  A window presented in the figure 16 (a,b) displays the characteristics of each UAVs, indicating its name, IoT sensor, the energy that decreases according to the model represented in the implementation part, and the variation of the time required to reach the point of arrival.

  During their mobility, drones collect the desired data in the defined periods.



(a)

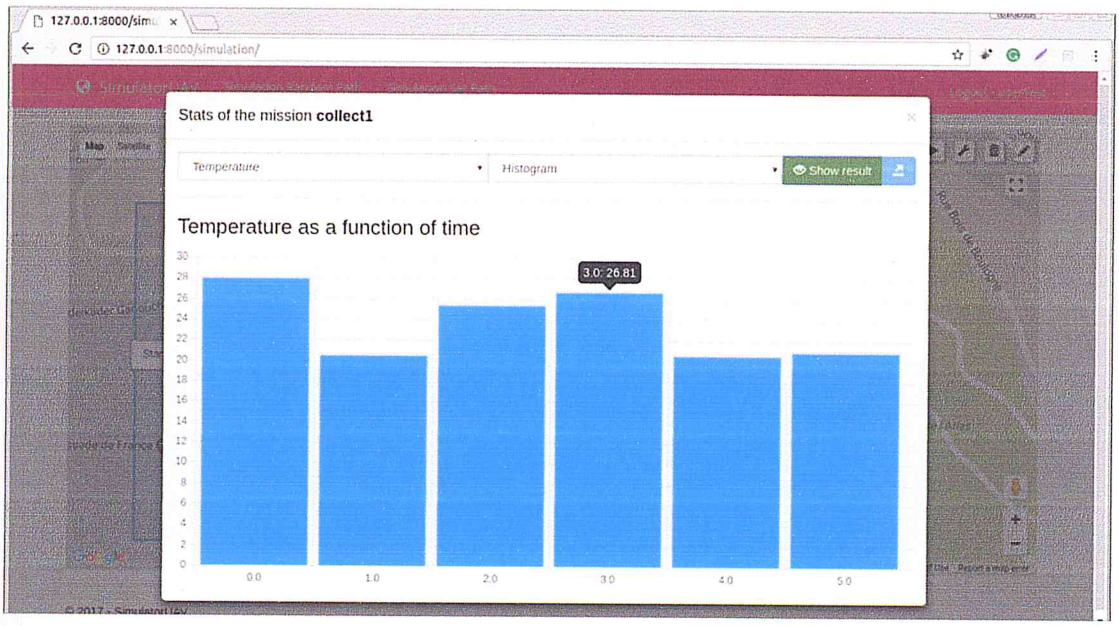


(b)

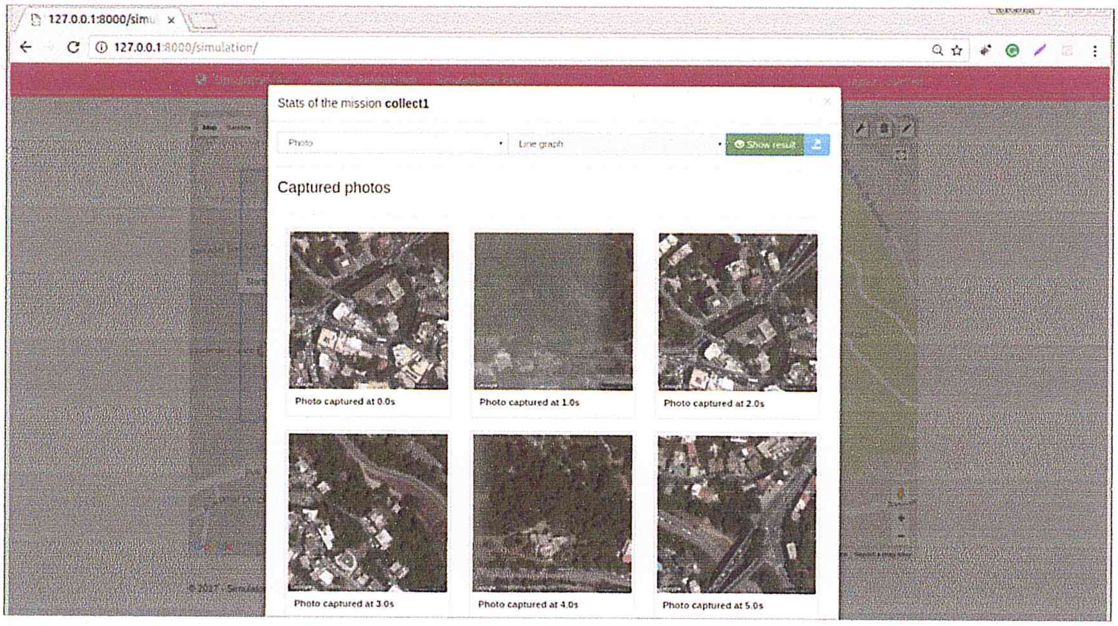
Figure 16: Display Window for UAVs mobility

- Displaying and exporting results:

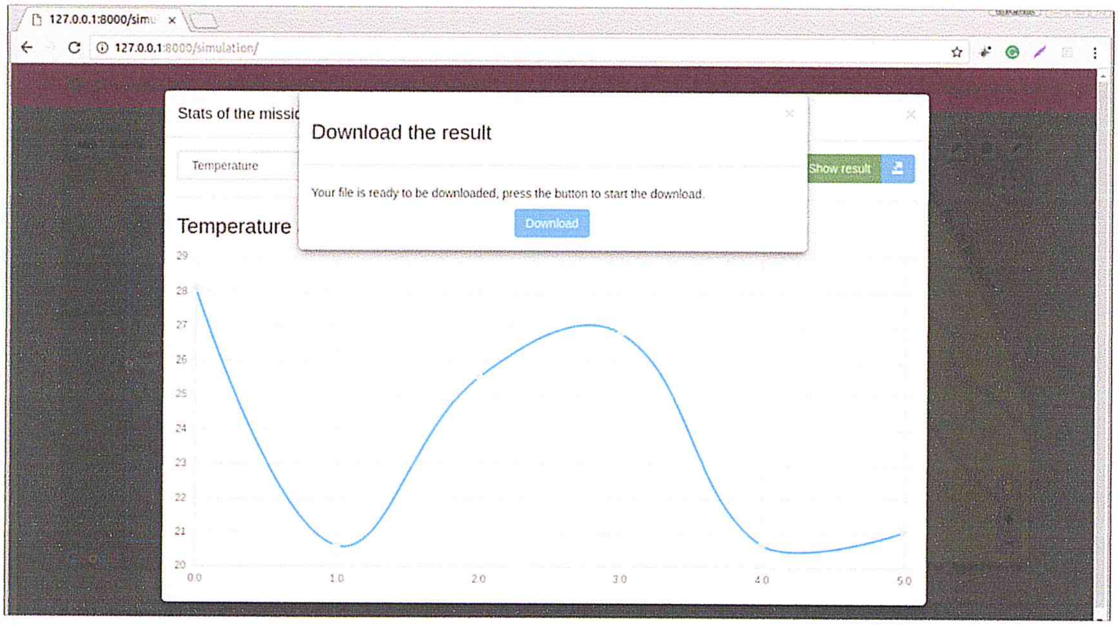
  At the end of the simulation, we can consult the data collected in the graphs (Histogram, Pie, and Line), as well as the total energy and time of the mission. The figure 17 (a, b and c) presents the display window for displaying data on a graph.



(a)



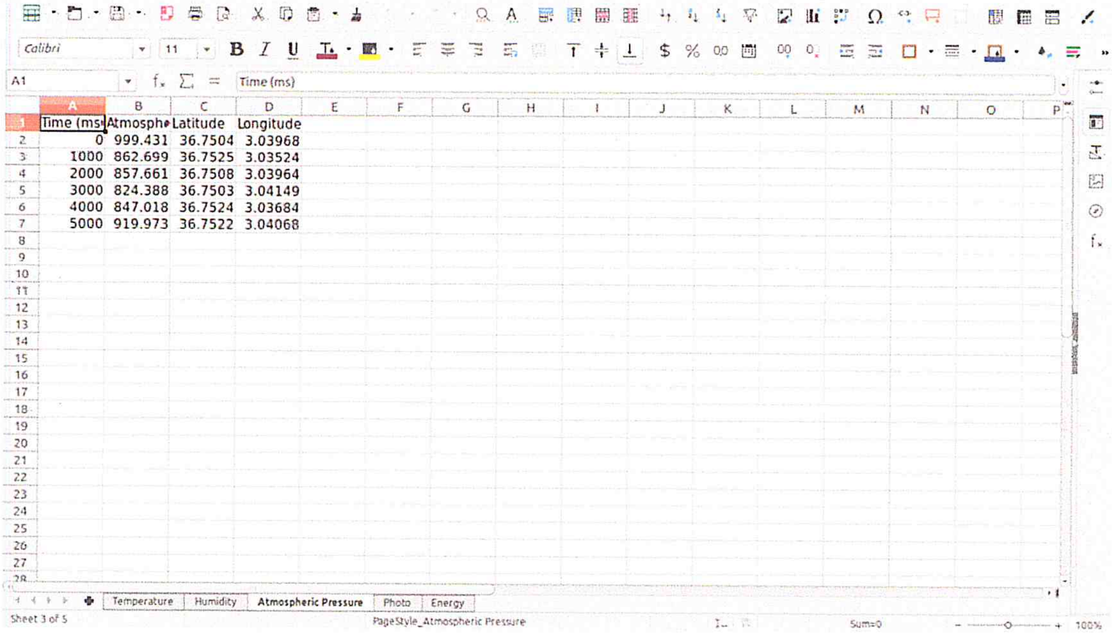
(b)



(c)

Figure 17: Display Window for Displaying data on a graph

For future needs, the results are recorded in an excel file, which can be downloadable. The file contains the results of the data collected, as showing in the figure 18 (a).



(a)

Figure 18: Excel file for exporting data

4.8 Conclusion

In this last chapter, we have presented the different editors, programming languages, and frameworks that we used during the implementation of our system. We also cited a model of the UAV, the implementation of the various modules of the simulator, and the various exceptions dealt with during the development. In the end, we tested our simulator on a mission aimed to collect data and monitor an area by generating real photos.

General Conclusion

Unmanned aerial vehicles, or UAVs, have been used by militaries around the globe for a number of years and are now a key technology for intelligence, surveillance, and reconnaissance operations. To provide this services, UAVs are equipped with numerous Internet of Things devices. Their role is continually expanding where the necessity to design a simulator to reflect the real environment and test the different scenarios of UAVs mission. We started this work by presenting a state-of-the-art of three concepts: Internet of Things (IoT), Unmanned Aerial Vehicles (UAVs), and simulation. In this part, we began by presenting the concept of IoT, its definition, characteristics, the different technologies used, the communication models, a classification of the fields of application and the advantages and limitations of this domains. Then we presented the UAVs (drones), their components, a classification that is based on several parameters, their application, and the advantages and disadvantages. At the end of this chapter, introduced some concepts of simulation, its definition, the various uses, some examples of a simulator, and also the advantages and disadvantages.

In the second chapter, we first began by presenting the vocabulary related to mission planning, and some related constraints. The planning was then defined in the context of an unmanned aerial vehicle. Several researchers used different modeling for the problem of path planning. The diversity of the modeling is due to the complexity of the problem and to the nature of the mission (surveillance, mapping, etc.) and the external resources (UAVs, sensors, base station, etc.), which imposes the problem of satisfying these constraints.

We implemented a design for the desired simulator. From there, we have presented the assumptions of the system, the different functional and technical needs, then, we presented the global architecture of the system. This architecture has made it possible to break down the systems into 3 main modules: path planning, data collection, and communication, which we have detailed each module separately, citing the different scenarios for each one.

Finally, we presented the different technical choices of our implementation. Then, we have detailed the implementation of each module, giving some code presenting the main functions for each one. At the end of the chapter, a scenario of a UAV mission has been established and tested which reflects the behavior of drones in the real environment.

This work, which consists of the implementation and realization of a simulator for the drones based on the IoT concepts, allowed us to acquire new knowledge in several fields, such as, the Internet of Things and the UAV with their planning of a predefined mission. This study will serve us later as a basis for proposing a complete modeling of a mission for autonomous aerial vehicles.

The predefined objectives for our implementation are respected and all developed. Different perspectives for our system can be considered, such as:

- Implement the system on a 3D plane, instead of 2D. For this, we must take into account the elevation,
- Consider that the system takes into account the collision and avoidance of obstacles,
- Implement the real part, using real hardware like sensors, actuators, etc.

Appendix A

Project Management

Project management is the application of knowledge, skills, tools, and techniques to project activities to meet project requirements and objectives. It includes 3 keys features:

- Identifying what is needed or to be achieved (requirements).
- Addressing needs, concerns, and expectations.
- Balancing competing constraints (scope, quality, schedule, budget, resources, and risks).

Our master's project entitled "Unmanned Aerial Vehicles and Internet of Things Simulator" was carried out through a general study to familiarize with the notions of Internet of Things, Unmanned Aerial Vehicles, and simulation, as well as to have a general overview of the work that is done in these fields, and also to understand a mission of drones in order to properly handle our project. Then we opted for the design and realization of our simulator.

A.1 Methodology of adopted development

The realization of our project has required the adoption of a methodological approch of software development. Considering the size and the risks of the project, we have chosen to follow the development method in V. The V-model is a Software Development Life Cycle (SDLC) model where execution of processes happens in a sequential manner in a V-shape. It is also known as Verification and Validation model [119] . The V-Model is an extension of the waterfall model and is based on the association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle, there is a directly associated testing phase. This is a highly-disciplined model and the next phase starts only after completion of the previous phase.

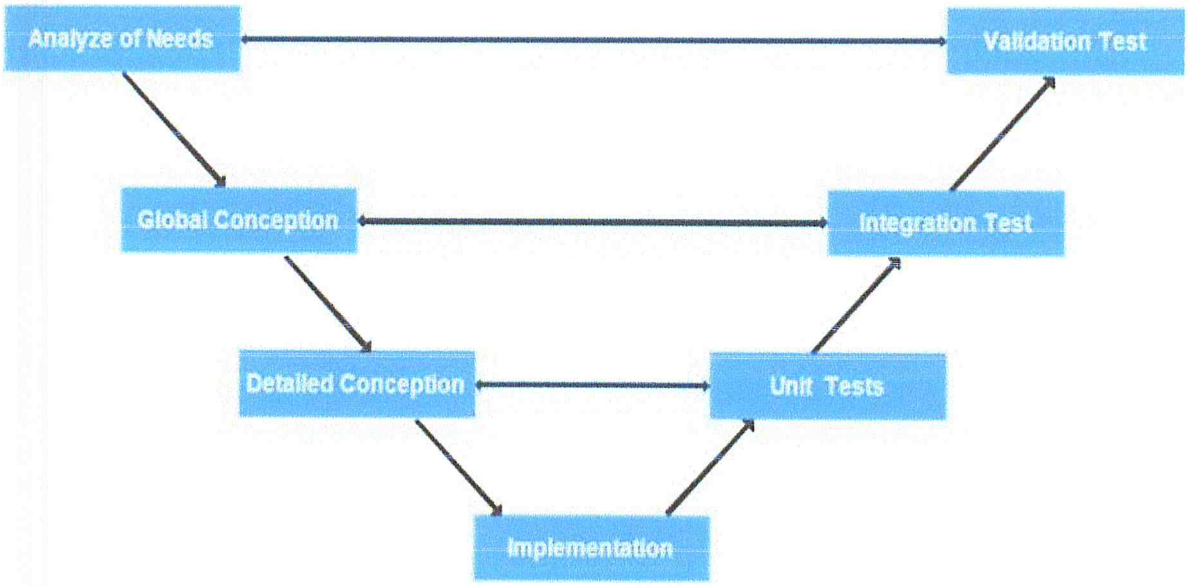


Figure 19: V life cycle model

Under the V-Model, the corresponding testing phase of the development phase is planned in parallel. So, there are Verification phases on one side of the 'V' and Validation phases on the other side. The implementation Phase joins the two sides of the V-Model.

A.1.1 Analyze of Needs

The first phase of development following cycle V is called "Analyze of Needs" phase. This phase consists of describing the functional requirements as well as possible in order to get the full response of the developed system. It often starts from a specification document drawn up by the client, describing his expectations. The development team then uses this document to analyze these needs and re-express them in more appropriate terms. During this step, we also defined specific requests, such as compliance with certain graphics standards, response times, hardware on which the software should work, etc. This step results in a document of specifications - functional and technical - which can be schematized by diagrams and histograms which can be documented. The test phase corresponding to analyze of needs is the validation test phase.

A.1.2 Global Conception

The global conception phase describes the overall architecture of the system. It defines the components of the system and how they communicate to perform system functions. It is an important step in the life cycle because it's at this point that the development team based on his modular decomposition of the problem in order to be able to reply for the requirements. It represents the first brick in the solution.

A.1.3 Detailed Conception

In this step, each module presented in the overall design is detailed. The functional aspects are highlighted and the algorithms and business processes are described more precisely. Integration testing verifies the deliverables of both conceptions steps.

A.1.4 Implementation

This is the lowest level realization step. It consists in translating the detailed design into a given development language. It is also at this level that the considerations related to the choices of the material, the processes programming and the algorithms of operation are introduced. The tests corresponding to this phase are the unit tests.

A.1.5 Unit Test

Unit tests designed in the module design phase are executed on the code during this validation phase. Unit testing is the testing at the code level and helps eliminate bugs at an early stage, though all defects cannot be uncovered by unit testing.

A.1.6 Integration Test

Integration testing is associated with the architectural design phase. Integration tests are performed to test the coexistence and communication of the internal modules within the system.

A.1.7 Validation Test

Validation test is associated with the business requirement analysis phase and involves testing the product in the user environment. It tests uncover the compatibility issues with the other systems available in the user environment. It also discovers the non-functional issues such as load and performance defects in the actual user environment.

A.2 Choice of Methodology

The criteria that motivate the choice of our project management methodology (the V method) is the size of the project and its nature. Indeed, the V method is adaptable, which makes the classical methodologies efficient enough to carry out the project realization. In addition, the project does not involve large risks and the objectives are clear. Therefore, the V-model is the most suitable for our case study, because it allows highlighting the specifications of the system and emphasizes the phase of the tests to guarantee a good quality of the final product.

A.3 Project Management Tools

To carry out this project, we have been helped by open source tools, we mention:

- Microsoft Office Project, for the provisional planning.
- Latex, it allows writing quality documents.
- Gmail, Skype, and Dropbox to communicate between the different actors involved in this project and to exchange the different deliverables.

Bibliography

[1]: **D. Nield**, "Boeing's latest patent reveals a drone that can transform into a submarine". ScienceAlert. [Online]. Available: http://www.sciencealert.com/boeing-s-latest-patent-reveals-a-drone-that-can-transform-into-a-submarine [Accessed: 07-Mar-2016].

[2]: "Information and Communications in Japan". Economic Research Office, Ministry of Internal Affairs and Communications(MIC),Japan,2011. [Online]. Available: http://www.soumu.go.jp/johotsusintokei/whitepaper/eng/WP2011/2011-index.html [Accessed: 11-Mar-2016]

[3]: **J. Siminski**, "Fukushima plant's radiation levels monitored with an UAV". The Aviationist, 29-Jan-2014, [Online]. Available: https://theaviationist.com/2014/01/29/fukushima-japan-uav/ [Accessed:12-Jun-2016].

[4]: **E. Mavropoulou**, "The success of the Migrant Offshore Aid Station (MOAS)". Human Rights At Sea. [Online]. Available: https://www.humanrightsatsea.org/the-success-of-the-migrant-offshore-aid-station-moas/ [Accessed:12-Jun-2016].

[5]: **A. Qiantori, A. B. Sutiono, H. Hariyanto, H. Suwa, and T. Ohta**, "An Emergency Medical Communications System by Low Altitude Platform at the Early Stages of a Natural Disaster in Indonesia". J. Med. Syst., vol. 36, no. 1, pp. 41–52, Mar. 2010.

[6]: **J. Rogers**, "How Drones Are Helping The Nepal Earthquake Relief Effort". FoxNews.com, 30-Apr-2015. [Online]. Available: http://www.foxnews.com/tech/2015/04/30/how-drones-are-helping-nepal-earthquake-relief-effort.html [Accessed: 07-Mar-2016].

[7]: "History – Aalto University". [Online]. Available: http://www.aalto.fi/en/about/history/ [Accessed: 18-Jul-2017].

[8]: "Organization – Aalto University". [Online]. Available: http://www.aalto.fi/en/about/organization/ [Accessed: 18-Jul-2017].

[9]: "Department of Communications and Networking – Aalto University". [Online]. Available: http://comnet.aalto.fi/en/ [Accessed: 18-Jul-2017].

[10]: "Aalto MOSAIC Lab Home Page". [Online]. Available: http://www.mosaic-lab.org/default.aspx [Accessed: 19-Jul-2017].

[11]: "History – CERIST". [Online]. Available: http://www.cerist.dz/index.php/en/about-us-en-3/730-historique [Accessed: 19-Jul-2017].

[12]: "Research and Development Activities". CERIST Journal.

[13]: "Wireless Sensors Networks & Applications Group". [Online]. Available: http://wsn.cerist.dz/ [Accessed: 20-Jul-2017].

[14]: "Overview of the Internet of Things". Recommendation ITU-T Y.2060, 15 June 2012. [Online]. Available: http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=Y.2060 [Accessed: 20-Mar-2017].

[15]: **Fabien Vannel, and Nabil Abdennadher.** "*Introduction to IOT*". Genève : HEPIA. [Online]. Available: http://wsn.cerist.dz/ [Accessed: 20-Jul-2017].

[16]: **Keyur K Patel, and Sunil M Patel.** "Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges". International Journal of Engineering Science and Computing IJESC. Volume 6. Issue No. 5, 2016.

[17]: **Dr. Ovidiu Vermesan SINTEF, Norway, Dr. Peter FriessEU, and Belgium.** "Internet of Things–From Research and Innovation to Market Deployment". s.l. : River publishser's series in communications, 2014.

[18]: **K. R. Rao, Zoran S. Bojkovic, and Dragorad A. Milovanovic,** . "Wireless Multimedia communications: Convergence, DSP, QoS, and Security". Chapter 5: Wireless Networking Standard (WLAN, WPAN, WMAN, WWAN). CRC Press. Taylor & Francis Group 2008. Pages 99-166.

[19]: **Ing. Milan Šimek, Ing. Ivan Míča, Ing. Jan Kacálek, Ing. Radim Burget**. "Bandwidth Efficiency of Wireless Networks of WPAN, WLAN, WMAN and WWAN". Electro Revue ISSN 1213-1539. 28.08.2007.

[20]: **Pankaj Barodia.** "Routing Protocols in Wireless Network". Deenbandhu Chhotu Ram University of Science and Technology. Murthal, Sonipat (Haryana), June 2011.

[21]: **Jin-Shyan Lee, Yu-Wei Su, and Chung-Chou Shen.** "A Comparative Study of Wireless Protocols: Bluetooth, UWB, Zigbee, and Wi-Fi". The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON). November. 5-8, 2007. Taipei, Taiwan.

[22]: **BRIAN RAY.** "Bluetooth Vs. Bluetooth Low Energy: What's The Difference?". November 01, 2015. [Online]. Available: https://www.link-labs.com/blog/bluetooth-vs-bluetooth-low-energy [Accessed: 31-Aug-2017].

[23]: **D. Porcino and W. Hirt**, "Ultra-wideband radio technology: Potential and challenges ahead". IEEE Commun. Mag., vol. 41, no. 7, pp. 66-74, July 2003.

[24]: **Jin-Shyan Lee and Yuan-Ming Wang**. "Experimental Evaluation of ZigBee-Based Wireless Networks in Indoor Environments". Journal of Engineering. Volume 2013. Article ID 286367, 9 pages.

[25]: **John D'Ambrosia**. "IEEE 802 LAN/MAN Standards Committee, LMSC, LAN/MAN Standard Committeee (Project 802)". [Online]. Available: http://www.ieee802.org/ [Accessed: 9-Aug-2017].

[26]: **Sourangsu Banerji and Rahul Singha Chowdhury**. "Wi-Fi & WiMAX: A Comparative Study". Original Publication: Indian Journal of Engineering, Vol.2, Issue. 5, 2013.

[27]: GSM - About Us. [Online]. Available: https://www.gsma.com/aboutus/gsm-technology/gsm [Accessed: 9-Aug-2017].

[28]: **Inanoglu, Hakan, John Reece, and Murat Bilgic**. "General Packet Radio Service (GPRS)". Wireless Network Deployments (2002): 197-213.

[29]: **Arefin and Ashraful.** "UNIVERSAL MOBILE TELECOMMUNICATION SYSTEM". (2013). Department of Electronics & Communication Engineering NORTHERN UNIVERSITY BANGLADESH.

[30]: **X.L. Jia, Q.Y. Feng, and C.Z. Ma.** "An efficient anti-collision protocol for RFID tag identification". IEEE Communications Letters, vol.14, no.11, pp.1014-1016, 2010.

[31]: **K. Finkenzeller.** "RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication". New York: Wiley, 2010.

[32]: K. Ok and *al.*, "Current Benefits and Future Directions of NFC Services". Intl. Conf. on Education and Management Technology ICEMT, 334 (2010).

[33]: **Tschofenig. H, et al.** *"Architectural Considerations in Smart Object Networking"*. s.l. : Internet Architecture Board, March 2015. Tech. no. RFC 7452.

[34]: "Meet the Nest Thermostat | Nest.". Nest Labs. [Online]. Available: https://nest.com/thermostat/meet-nest-thermostat/ [Accessed: 6-Aug-2017].

[35]: "Samsung Privacy Policy -- SmartTV Supplement.". Samsung Corp. [Online]. Available: http://www.samsung.com/sg/info/privacy/smarttv/ [Accessed: 6-Aug-2017].

[36]: "Internet of Things". Pan European Research and Innovation Vision, European Research Cluster on the Internet of Things. October 2011. [Online]. Available: http://www.internet-of-things-research.eu/pdf/IERC_IoT-Pan%20European%20Research%20and%20Innovation%20Vision_2011_web.pdf [Accessed: 6-Aug-2017]

[37]: **Gubbi Jayavardhana, Buyya Rajkumar, Marusic Slaven, and Palaniswami Marimuthu.**: "Internet of Things (IoT): A vision, architectural elements, and future directions". Future Generation Computer Systems (ELSEVIER Journal). 2013.

[38]: **Andrea Zanella and Lorenzo Vangelista.** "Internet of Things for Smart Cities". IEEE INTERNET OF THINGS JOURNAL, Vol. 1, NO. 1, February 2014.

[39]: **Leonardo Barreto Campos, Carlos Eduardo Cugnasca, André Riyuiti Hirakawa and José Sidnei C. Martini.** "Towards an IoT-based System for Smart City". 2016 IEEE International Symposium on Consumer Electronics.

[40]: **Rosario Miceli.** "Energy Management and Smart Grids" .Energies2013, ISSN 1996-1073. Published 22 April 2013.

[41]: **Mobyen Uddin Ahmed, Mats Björkman, Aida Čaušević, Hossein Fotouhi, and Maria Lindén.** "An Overview on the Internet of Things for Health Monitoring Systems". Conference: 2nd EAI International Conference on IoT Technologies for HealthCare, At Rom. October 2015

[42]: **Ron Davies.** "The Internet of Things Opportunities and challenges". EPRS | European Parliamentary Research Service. May 2015. [Online]. Available: http://www.europarl.europa.eu/RegData/etudes/BRIE/2015/557012/EPRS_BRI(2015)557012_EN.pdf [Accessed: 8-Aug-2017]

[43]: **Friedemann Mattern and Christian Floerkemeier.** "From the Internet of Computers to the Internet of Things". [Online]. Available: https://www.vs.inf.ethz.ch/publ/papers/Internet-of-things.pdf [Accessed: 10-Aug-2017]

[44]: **Louis COETZEE and Johan EKSTEEN.** "The Internet of Things- Promise for the future? An introduction ". IST-Africa 2011 Conference Proceedings. Paul Cunningham and Miriam Cunningham (Eds). IIMC International Information Management Corporation, 2011. ISBN: 978-1-905824-26-7

[45]: "Internet of Things: Privacy & Security in a Connected World". s.l. : FTC Sta Report, January 2015.

[46]: "The Internet of Things: Seizing the benefits and addressing the challenges". Working Party on Communication Infrastructure and Services Policy. Directorate for Science, Technology and Innovation Committee on Digital Economy Policy. 24 May 2016.

[47]: **HaiYang Chao, YongCan Cao, and YangQuan Chen** "Autopilots for Small Unmanned Aerial Vehicles: A Survey". International Journal of Control, Automation, and Systems (2010) 8(1):36-44.

[48]: **Eisenbeiss, Henri.** "*A MINI UNMANNED AERIAL VEHICLE (UAV): SYSTEM OVERVIEW.* PROCESSING AND VISUALIZATION USING HIGH-RESOLUTION IMAGERY". Pitsanulok, Thailand : s.n., November 2004. International Workshop.

[49]: "Unmanned Aerial Vehicles Block Diagram – Electronic products". 2013. [Online]. Available: https://www.electronicproducts.com/Electromechanical_Components/Motors_and_Controllers/Unmanned_Aerial_Vehicles_Block_Diagram.aspx [Accessed: 18-Aug-2017].

[50]: "Unmanned Systems Integrated Roadmap FY2011-2036". [Online]. Available: http://www.acq.osd.mil/sts/docs/Unmanned%20Systems%20Integrated%20Roadmap%20FY2011-2036.pdf [Accessed: 10-Aug-2017].

[51]: "Unmanned Systems Roadmap 2007-2032". [Online]. Available: http://www.globalsecurity.org/intell/library/reports/2007/dod-unmanned-systems-roadmap_2007-2032.pdf [Accessed: 10-Aug-2017].

[52]: **Kevin L. Digman, Major.** "UNMANNED AIRCRAFT SYSTEMS IN A FORWARD AIR CONTROLLER (AIRBORNE) ROLE". USMC, Maxwell Air Force Base, Alabama April 2009.

[53]: "UAS ROADMAP 2005- 2030". [Online]. Available: https://fas.org/irp/program/collect/uav_roadmap2005.pdf [Accessed: 11-Aug-2017].

[54]: **Javier Bilbao, Andoni Olozaga, Eugenio Bravo, Olatz García, Concepción Varela and Miguel Rodríguez.** "How design an unmanned aerial vehicle with great efficiency in the use of existing resources". INTERNATIONAL JOURNAL OF COMPUTERS Issue 4, Volume 2, 2008.

[55]: **BENTO, MARIA DE FATIMA.** "*Unmanned Aerial Vehicles: An Overview*". s.l. : Inside GNNS, 2008.

[56]: **J. Everaerts.** "THE USE OF UNMANNED AERIAL VEHICLES (UAVS) FOR REMOTE SENSING AND MAPPING". The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B1. Beijing 2008.

[57]: **Jack Brown.** "DRONE USES: THE AWESOME BENEFITS OF DRONE TECHNOLOGY". [Online]. Available: http://mydronelab.com/blog/drone-uses.html [Accessed: 11-Aug-2017]

[58]: **Mark Thompson, Bobby Ghosh.** *The CIA's Silent War in Pakistan.* Washington : TIME, 1 June 2009.

[59]: *Airborne Bomb Detection.* s.l. : HSS Development, 2016.

[60]: **George Friedrich**. "Applications of military and non-military Unmanned Aircraft Systems (UAV)". University of Applied Sciences Stralsund, Current Trends in Science and Technology.

[61]: **Dr Alexander V Koldaev**. "NON-MILITARY UAV APPLICATIONS". February 2007.

[62]: **T. Tanzi, L. Apvrille, J. L. Dugelay, and Y. Roudier.** *"UAVs for humanitarian missions: Autonomy and reliability".* s.l. : IEEE Global Humanitarian Technology Conference (GHTC), 2014. pp. 271-278.

[63]: **Naser Hossein Motlagh, Tarik Taleb, and Osama Arouk.** *"Low-Altitude Unmanned Aerial Vehicles-Based Internet of Things Services: Comprehensive Survey and Future Perspectives".* s.l. : IEEE Internet of Things Journal, 2016.

[64]: **Bobbe, Tom.** "The role of UAV technology in Forest Service natural resource management". Center Salt Lake City, Utah : USDA Forest Service Remote Sensing Applications , July 2005.

[65]:] **Coifman, B., McCord, M., Mishalani, M., Redmill, K.**, "Surface Transportation Surveillance from Unmanned Aerial Vehicles". Proc. of the 83rd Annual Meeting of the Transportation Research Board, 2004.

[66]: **Evan Baldwin Carr.** "Unmanned Aerial Vehicles: Examining the Safety, Security, Privacy and Regulatory Issues of Integration into U.S. Airspace".

[67]: **María Fernández Raga, Covadonga Palencia, and Enoc Sanz Ablanedo**. "Advantages and disadvantages with drones in determining the erosion of a fire zone". Geophysical Research Abstracts. Vol. 18, EGU2016-704-1, 2016. EGU General Assembly 2016.

[68]: **Elena Fabiola Ruiz Ledesma, Juan Jesús Gutiérrez García.** "Simulation as a resource in the Calculus Solving". 2013.

[69]: **Nigel Gilbert and Klaus G**. Troitzsch. "Simulation for the Social Scientist". Open University Press. 1999.

[70]: **Stewart Robinson.** "Simulation: The Practice of Model Development and Use". Warwick Business School.

[71]: "List of computer software – Wikipedia". [Online]. Available: https://en.wikipedia.org/wiki/List_of_computer_simulation_software [Accessed: 15-Aug-2017].

[72]: **fpvsimulators.** "Aerofly RC 7 | Drone Simulator Review". January 16, 2016. [Online]. Available: http://fpvsimulators.com/drone-simulator-aerofly-rc-7-review/ [Accessed: 02-Sep-2017].

[73]: **Víctor Rodríguez Fernandez.** "Development of a Multi-UAV Simulator to Analyze the Behavior of Operators in Coastal Surveillance Missions". Polytechnic Higher School. Department of Computer Engineering. April 2015.

[74]: **ArduPilot Dev Team.** "Mission Planner Overview – Mission Planner documentation". [Online]. Available: http://ardupilot.org/planner/docs/mission-planner-overview.html [Accessed: 02-Sep-2017].

[75]: **Cat Ellis.** "The best free drone simulator 2017 | TechRadar". [Online]. Available: http://www.techradar.com/news/the-best-free-drone-simulator [Accessed: 02-Sep-2017].

[76]: **Sudarshan Kumar Patel.** "Simulation". National Academy of Agricultural Research Management. 2013.

[77]: **Tanil Cagatay, Chirag Warty and Esam Obiedat.** "Collaborative mission planning for UAV cluster to optimize relay distance". In : IEEE Aerospace Conference Proceedings. 2013. issn : 1095323X. doi : 10.1109/AERO.2013.6497407.

[78]: "WHAT IS AN AUTONOMOUS SYSTEM? - Abu Dhabi Autonomous Systems Investments Co LLC". [Online]. Available: http://adasi.ae/autonomous-systems/what-is-an-autonomous-system.aspx [Accessed: 26-Aug-2017].

[79]: **Chandler, Phillip R., Meir Pachter, and Steven Rasmussen.** "UAV cooperative control". American Control Conference, 2001. Proceedings of the 2001. Vol. 1. IEEE, 2001.

[80]: **Lamont, Gary B., James N. Slear, and Kenneth Melendez.** "UAV swarm mission planning and routing using multi-objective evolutionary algorithms". Computational Intelligence in Multicriteria Decision Making, IEEE Symposium on. IEEE, 2007.

[81]: **Evsen Yanmaz, and al.** "On Path Planning Strategies for Networked Unmanned Aerial Vehicles". IEEE INFOCOM 2011 Workshop on M2MCN-2011.

[82]: **Paull, Liam, and al.** "Sensor-driven online coverage planning for autonomous underwater vehicles". IEEE/ASME Transactions on Mechatronics 18.6 (2013): 1827-1838.

[83]: **Yani Cui, Jia Ren, Wencai Du, and Jingguo Dai.** "UAV target tracking algorithm based on task allocation consensus". Journal of Systems Engineering and Electronics. Vol. 27, No. 6, December 2016, pp.1207 –1218.

[84]: **Julien Guitton, and Jean-Loup Farges.** "Planification de mission pour un ensemble de drones de combat aériens ". JFPDA 2007.

[85]: **BEARD R. W., MCLAIN T., GOODRICH M. & ANDERSON E.** "Coordinated target assignment and intercept for unmanned air vehicles". In IEEE transactions on robotics and automation, 2002. Volume 18, p. 911–922.

[86]: **KUWATA Y.** "Real-time trajectory design for UAVs using receding horizon control". 2003. Master's thesis, Dept. of Aeronautics and Astronautics, MIT.

[87]: **Cristian Oliver Ramırez Atencia.** "Modelling Unmanned Vehicles Mission Planning problems as Constraint Satisfaction Problems". September 2014.

[88]: **Moreno Cristian Ramirez atencia, Gema Bello orgaz María D and David Camacho.** "Solving complex multi-UAV mission planning problems using multi-objective genetic algorithms". 2016. Soft Computing.

[89]: **Jiang Jun, and al.** "Expert Systems with Applications Vehicle routing problem with a heterogeneous fleet and time windows". 2014. In : EXPERT SYSTEMS WITH APPLICATIONS 41.8, p. 3748–3760.

[90]: **Fogel, D. B.** "An Evolutionary Approach to the Traveling Salesman Problem". Biological Cybernetics. December 1988. Volume 60, Issue 2, pp 139–144.

[91]: **P. Larrañaga, and al.** "Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators ". Artificial Intelligence Review. December April 1999, Volume 13, Issue 2, pp 129–170.

[92]: **Casquilho, Miguel AS.** "Travelling Salesman Problem". 2012. Technical University of Lisbon, Ave. Rovisco Pais, 1049-001 Lisboa, Portugal.

[93]: **Haque, Mohd Junedul, and Khalid W. Magld.** "Improving the Solution of Traveling Salesman Problem Using Genetic, Memetic Algorithm and Edge assembly Crossover". 2012. International Journal of Advanced Computer Science and Applications (IJACSA). Vol. 3, No. 7.

[94]: **Tonci Caric and Hrvoje Gold.** "Vehicle Routing Problem". In-teh. Croatia, 2008.

[95]: **Suresh Nanda Kumar, Ramasamy Panneerselvam.** "A Survey on the Vehicle Routing Problem and Its Variants ". Intelligent Information Management, 2012. Vol 4, p 66-74.

[96]: **K.C. Tan, and al**. "Heuristic methods for vehicle routing problem with time window". Artificial Intelligence in Engineering. 2001. p 281-295.

[97]: **F. Guerriero, and al.** "A Multi-objective Approach for Unmanned Aerial Vehicle Routing Problem with Soft Time Windows Constraints". Applied Mathematical Modelling, Elsevier, 2014, 38 (3), p.839-852.

[98]: **Jun Jiang**. "Vehicle routing problem with a heterogeneous fleet and time windows". Journal Expert Systems with Applications. Volume 41 Issue 8, June 2014. Pages 3748-3760.

[99]: **Palo Xerox and Hill Road**. "Dynamic Constraint". System Sciences Laboratory. Xerox Palo Alto Research Center. 1990. p. 25–32.

[100]: **M. K. Simon and M.-S. Alouini.** "Digital Communication over Fading Channels". 2nd ed. New Jersey, USA: John Wiley & Sons, 2005.

[101]: **A. F. Molisch.** "Wireless Communications". Wiley-IEEE Press, 2005.

[102]: "PyCharm: Python IDE for Professional Developers by JetBrains". [Online]. Available: https://www.jetbrains.com/pycharm/ [Accessed: 23-Aug-2017].

[103] "Sublime Text - Wikipedia". [Online]. Available: https://en.wikipedia.org/wiki/Sublime_Text [Accessed: 23-Aug-2017].

[104]: **Van Rossum, Guido.** "Python Programming Language". USENIX Annual Technical Conference. Vol. 41. 2007.

[105]: **Flanagan, David.** "JavaScript: the definitive guide". O'Reilly Media, Inc., 2006.)

[106]: "Introduction to jQuery". [Online]. Available: https://openclassrooms.com/courses/intro-to-jquery [Accessed: 23-Aug-2017].

[107]: **Bierman, Gavin, Martín Abadi, and Mads Torgersen.** "Understanding typescript". European Conference on Object-Oriented Programming. Springer, Berlin, Heidelberg, 2014.

[108]: "The Web framework for perfectionists with deadlines - Django". [Online]. Available: https://www.djangoproject.com/ [Accessed: 23-Aug-2017].

[109]: "Django Overview". [Online]. Available: https://www.tutorialspoint.com/django/django_overview.htm [Accessed: 23-Aug-2017].

[110]: "Google Maps". [Online]. Available: maps.google.com [Accessed: 28-Aug-2017].

[111]: **Gabriel Svennerberg.** "Google Maps API 3". Apress. 2010.

[112]: "Google Maps JavaScript API – Google Developers". [Online]. Available: https://developers.google.com/maps/documentation/javascript/?hl=fr [Accessed: 24-Aug-2017].

[113]: "REST API - ServiceNow Wiki". [Online]. Available: http://wiki.servicenow.com/index.php?title=REST_API#gsc.tab=0 [Accessed: 24-Aug-2017].

[114]: **Margaret Rouse.** "What is RESTful API? - Definition from WhatIs.com". [Online]. Available: http://searchcloudstorage.techtarget.com/definition/RESTful-API [Accessed: 24-Aug-2017].

[115]: **Margaret Rouse.** "What is RESTful API? - Definition from WhatIs.com". [Online]. Available: http://searchcloudstorage.techtarget.com/definition/RESTful-API [Accessed: 24-Aug-2017].

[116]: **A. R. Mohammad and D. Simon.** "Energy-Efficient Sensing in Wireless Sensor Networks Using Compressed Sensing". Sensors, vol. 14, no. 2, pp. 2822–2859, Feb. 2014.

[117]: **W. Chan, T.-C. Lu, and R.-J. Chen**. "Pollaczek-Khinchin formula for the M/G/1 queue in discrete time with vacations". IEE Proceedings - Computers and Digital Techniques, vol. 144, no. 4,pp. 222–226, Jul. 1997.

[118]: **A. Chelli, E. Zedini, M.-S. Alouini, J. R. Barry, and M. Patzold.** "Performance and delay analysis of hybrid ARQ with incremental redundancy over double Rayleigh fading channels". IEEE Trans. on Wireless Commun., vol. 13, no. 11, pp. 6245–6258, Nov. 2014.

[119]: **TutorialsPoint.** "SDLC". [Online]. Available:
https://www.tutorialspoint.com/sdlc/sdlc_tutorial.pdf [Accessed: 02-Sep-2017].

جامعة البليدة - 1
مكتبة
الكلية
كلية العلوم