UAV Trajectory Optimisation in Smart Cities using Modified A* Algorithm Combined with Haversine and Vincenty Formulas

Andreas Andreou, *Student Member*, *IEEE*, Constandinos X. Mavromoustakis, *Senior Member*, *IEEE*, Jordi Mongay Batalla, *Member*, *IEEE*, Evangelos K. Markakis, *Member*, *IEEE*, George Mastorakis, *Member*, *IEEE*, and Shahid Mumtaz, *Senior Member*, *IEEE*

Abstract— It is anticipated that the backbone of Smart Cities concerning automation and networking will be formed by Unmanned Aerial Vehicles in the imminent future. Therefore, our research focuses on developing advanced microcontrollers embedded with Artificial Intelligence techniques for self-governing Unmanned Aerial Vehicles. The main objective of this research was to enable full automation for the execution of flight paths with non-trivial sequences that will be performed with centimetre-level accuracy. Also, by utilising dynamic flight plans and trajectories, we aim to secure autonomous aviation based on norms, with control loops and fundamental constraints. More specifically, we evolved a novel algorithmic technique for trajectory optimisation, which deploys a modification to the A* search algorithm, implemented by the Haversine formula and enhances accuracy using Vincenty's formula. Furthermore, realistic values for trajectory optimisation and obstacle avoidance were found through the implementation of a simulative investigation. The outcomes of our methodology indicate that the safety constraints associated with the integration of Unmanned Aerial Vehicles in the urban environment can be significantly mitigated. Consequently, their effectiveness will be increased in realising their diverse operations and capabilities.

Index Terms— Unmanned aerial vehicles, air traffic control, air transportation, Smart vehicles, aircraft navigation, Internet of Things, Autonomous

INTRODUCTION

AVS are a prerequisite for the technological advancement of the mobility arena in smart cities. The future evolution of sixth-generation (6G) wireless networks is a source of significant excitement in the upcoming intelligent ecosystems, especially in intelligent transportation within urban environments. The ultimate objective of 6G is to enable efficient connectivity for massive-scale Internet of Things (IoT) networking. Therefore, the take-off for flying intercity vehicles is desirable with recent advances in lithium-ion batteries.

Developing innovative transportation solutions to address traffic problems is paramount for constructing sustainable smart cities. In conjunction with the IoT network system, the near-ground transportation infrastructure has become inextricably linked with the creation of smart cities [1]. As the name implies, smart cities are structured atop more intelligent data. It is a significant challenge because data is frequently an afterthought when creating new processes and opportunities. Therefore, using the right building blocks from the outset is vital, aligned with clear and compelling guidelines about best practices. To achieve scale, we need standards to qualify how we view and measure the world around us and how this data informs our decision-making processes.

Current transportation systems utilise ground-level, waterborne, high-altitude, and underground spaces, such as rapid transit systems, which are the most efficient in densely urbanised environments. Although, the immediate ground solution for transportation seems to be more suitable as the cost for the development will be low, in contradiction to an underground infrastructure, which requires a multimillion budget to be constructed. There are many advantages to using air mobility in urban environments, such as flexibility in transportation as discrete paths could be followed. However, to securely integrate this transportation system into the existing mobility landscape, we must consider several aspects and overcome obstacles hindering growth. Automated driving strategies have been advocated as significant advancements to ensure safe air transportation [2]. Safety is paramount, considering that aviation will levitate above residential areas. In this regard, safety integrity must place high targets throughout the development and use cycle. Considering the lack of comprehensive literature on safe ground-based aviation for UAVs, this study aims to set a milestone for activating the infrastructure that will integrate UAVs into future smart cities.

Therefore, we study the problems that arise during unmanned air traffic systems and aim to approach their solution through this research work. Our main goal will be to propose the development of software integrated into the intelligent system of the UAV that will launch and control the drone, which will simulate the flying vehicle, considering the actual situation on the ground. The objectives of the software will be to solve the following main tasks:

1. Automatic routing on actual terrain from one point to another, considering no-fly zones along the shortest route

2. Automatic linking of the map to the actual position of the operator using the GPS/GLONASS receiver

3. Calculation of the trajectory of the drone and the control signals necessary for its control

4. Two-way communication with the drone to transmit traffic control signals along the route with an adjustment based on tracking its position in real-time

The perspective of this project is that it consists of a blueprint of a small part of a more extensive system. We envision the system building paths according to the no-flight zones and extending to a 3D space with multiple echelons. It will also be able to simulate a flight of an actual transport and work with various simulations simultaneously. Hence, it could be tested by creating randomly generated simulated UAVs with random destinations. The program will sort out the paths depending on their length and approximate time of approaching a specific point on the track and choose the echelon they are flying through [3]. In addition, it will monitor and correct the routes of many UAVs in real-time to ensure the safety of their passengers.

A. Related Works

Automated guided UAVs are currently being implemented in huge areas, such as rescue and adaptation of exploration platforms, due to their inherent characteristics of flexible mobility and simple deployment [4], [5]. In addition, multiple UAVs can create an automated network of guided vehicles that can be used to collect data and communicate with remote receivers. However, there are still some key challenges that need to be addressed. For example, they are based on battery capacity, limiting transmission distance. In addition, the communication quality of an individual vehicle is not guaranteed, as the transmission power is limited [6], [7].

Nevertheless, UAVs, also generally regarded as drones, have been envisioned as a promising technology in recent years [8]. Ground-controlled UAVs are considered aerial users with access to the cellular network from the sky. In addition, UAVs can be used as flight base stations or relays to assist wireless communications. They can receive data from the ground and transmit it to remote base stations to overcome transmission distance limitations [9]. The paper’s authors [10] proposed utilising a swarm of UAVs to acquire data from the WSN network [11] whenever the base station is nonfunctional. In addition, they studied the correlation between the parameters of the network, considering the constraint of avoiding collisions between the UAVs.

Charging the batteries of UAVs through Energy and Data Dispensers (EDD) devices was combined by addressing the assignment of the bandwidth and redefining the tridimensionality trajectory. However, connecting to EDDs is not yet in the plans, and the capacity may be limited as the dwell time in the recharging areas available for the UAV is limited. Therefore, it will be paramount for the UAVs to define an optimal movement trajectory between the EDDs. Based on the energy consumption, the research work [12] presented a novel approach to jointly optimise the 3D flight path and data collection schedule of a fixed-wing UAV for secure and energy-efficient data collection under an interception attack. The article [13], based on a mathematically oriented approach, presented a UAV object delivery scenario, where in the presence of multiple no-fly zones, a UAV starts from the departure point, collects the objects from ground users, and finally delivers the collected objects to the destination [14].

Determining the trajectory in real-time is one of the gaps that our research fulfils, based on the aforementioned related works. Also, the numerical results demonstrate the superiority of the proposed algorithm as the new proposed methodology outperforms the accuracy results for the current literature. Furthermore, in the example with the moving target for landing yields compared to the original A* algorithm and the bidirectional version, the computation time is reduced by 77.02% and 70.52%, respectively. Also, it is shown that the improved algorithm A* length is longer to generate the trajectory satisfying kinematic constraints.

B. Novelty

Prevailing risk-based methods elaborate the plan of UAVs trajectory on graph search algorithms such as the Dijkstra algorithm [15], heuristic A* algorithm [16], genetic algorithm [17], probabilistic roadmaps, and rapidly exploring random Trees. In addition, the ant colony optimisation algorithm [18] has an acceptable performance in UAV path planning problems. However, traditional algorithms do not consider a path's risk as encapsulating the distance in the heuristic function. Therefore, to address the limitations in the literature, as mentioned earlier, we aim to contribute by eliminating the risks in the current route planning for UAVs. Thus, we are focusing on modifying the A* algorithmic technique by elaborating on the haversine formula and Vincenty's formula. The novelty is the approach that accurately evaluates automated guided UAV trajectories.

C. Contribution

The proposed model's main contribution is the trajectory optimisation based on the modification we applied to the A* algorithmic technique by utilising the haversine formula and Vincenty's formula. Also, the comprehensive risk assessment implementation incorporates evaluation standards and sensitivity analysis on risk coefficients to quantify multiple risks, mainly in urban environments. The comparison with five modified A* algorithmic techniques consequent the optimisation in the trajectory planning considering several aspects such as cost-effective path in terms of time and risk limitation.

LITERATURE REVIEW

Recent research has focused on exploiting traffic congestion solutions and relieving pressure on the city's existingtransportation networks, using near-ground spaces to be used by flying cars. Therefore, engineers and researchers elaborate on the infrastructure development that will enable transportation and the unoccupied near-ground spaces [19]. However, there have not been any approaches regarding the flying car safety system and the infrastructure of the air routes or the taxiways, connecting runways with aprons and hangars to the ground.

Governments have developed air traffic systems to ensure the safety of integrating UAVs into the urban environment. Unmanned Aircraft Systems Traffic Management (UTM) was deployed in the US [20]. U-Space was in Europe. The Urban Traffic Management of Unmanned Aircraft Systems (uTM-UAS) in Singapore [21] and the UAV Operation and Management (UOM) in China. The authors of the research work [22] have presented the efforts taken by the Federal Aviation Administration (FAA) and the National Aeronautics and Space Administration (NASA) to develop a traffic management system for unmanned aircraft. It was mandatory, as the safe integration of unmanned aerial vehicle systems into the national airspace system of the United States has been a top priority for the FAA for several years. A comparison analysis was applied between the FAA Universal Transverse Mercator and the Concept of Operations for EuRopean UTM Systems (CORUS) project based on U-space and presented during the IEEE 39th Digital Avionics Systems Conference (DASC) [23].

Hitherto, there have been some research approaches exploring flying car infrastructure. For example, in [24], the authors studied the required legislation and policies to enable the sustainability of flying cars. More specifically, they elaborate on safety, navigation, and infrastructure. Also, recently, [25] contributed to the research on the same topic by analysing urban flying car scenarios and utilising an agent-based approach with various traffic conditions. Throughout [26], the authors studied the requirements for ensuring adequate law enforcement systems to ensure mandatory collision avoidance technology systems for reducing accidents to flying car passengers and people on the ground.

As the electric cars market continues to rise, hydrogen fuel cells which constitute an alternative type of vehicular fuel, come to the fore [27]. Moreover, hydrogen is the only fuel that could enable vertical take-off. Thus, according to Urban Aeronautics' founder, it will be the only viable way to allow the low-orbit hovering framework [28]. Alternatively, researchers exploit ways to reduce transportation energy consumption for ground vehicles. For instance, in [29], the authors had incorporated innovative panels between the wheels for energy metering purposes, intending to control fuel efficiency intelligently.

Leonhard Euler initially set the conjectures on spheroidal trigonometry [30] in 1755. While, Vincenty devised the algorithmic technique in [31] to computationally enable the solutions given by Legendre [32], Oriani [33], Bessel [34] and Helmert [35]. The research framework [16] focuses on integrating Helmert's methods into the current edge computing systems, achieving more efficiency in his results than Vincenty.

Recently, technological advancements allowed the highest level of energy density [36]. A wide range of composites, polymers, elastomers and vinyl, which are non-metallic materials and thus are lighter [37], enable the rise of autonomous control systems [38]. Therefore, mechanical and manufacturing engineering facilitates the efficient development of flying cars [39]. The primary motivation to date was the increased transportation issues, especially in urban environments and global economic growth. The objective of vertical take-off and landing operation mode and autonomous piloting for flying cars is to provide secure and sustainable transportation services for people and freights [40]. Furthermore, renewable energy sources [41] and autonomous piloting techniques [42] will be more efficient in intelligent urban ecosystems.

The need to solve the traffic problem in today's busy cities has accelerated the construction of a new transport system in smart cities. Therefore, start-up technology companies and those already established in the automotive and aircraft manufacturing sector have thoroughly elaborated on the design of effective flying car technologies. Hence, various flying cars are currently being manufactured and tested to be ready for mass production. The article [43] presented the effects of altitude and distance between aerial nodes on the recovery probability and verified them with simulations. In addition, the authors presented their testbed and preliminary experimental work showing promising results for aerial networks.

PROBLEM FORMULATION

Edsger Dijkstra 1959 introduced an algorithmic technique that can calculate the shortest path and is fundamental to solving path routing problems [44]. Briefly, through iterations, the algorithm chooses the nearest unselected vertices to an initially given vertex. After that, however, the path through the selected nodes should reach the destination. Several algorithm modifications have been applied since the initial approach; thus, we deployed an A* search algorithm reformation. The main goal of implementing the steps is to determine the least total value of the cost to travel along with the nodes. For example, the cost might be defined as the distance, time, or amount of data flow.

A. A* Search Algorithm

Navigation systems are generally outperformed by algorithms that can process graphical routes toward optimising the implementation process. The A* search algorithm constitutes one of the essential solutions for determining the optimal trajectory corresponding to the shortest path within any network [45]. Mathematically oriented A* aims to obtain the optimal route that minimises (1).

$$f(x)=c(x)+h(x)$$

(1)

Where x denotes the nodes, $c(x)$ the cost starts from the initial node and $h(x)$ denotes the problem-specific heuristic function that computes the narrowest path in terms of cost along the selected route. Since the heuristic function $h(x)$ is admissible, it ensures that the algorithmic technique generates the least-cost route. In addition, the heuristic function is called consistent if it satisfies (2) for every node with coordinates (x,y).

$$h(x)\leq d(x,y)+h(y)$$

(2)

Where the d(x,y) denotes the Euclidian distance from the node, as mentioned earlier. Hence, due to the consistent heuristic function h(x) the A* search algorithm determines the optimal path without processing any node more than once. The algorithm has the eligibility to conclude a solution if it exists on infinite graphs that satisfy (3) for the threshold ε.

$$d(x,y)>\varepsilon>0$$

(3)

From the threshold ε, we set the admissibility criterion, which assurances to compute the optimal trajectory and examine all the other equally notable trails. Nonetheless, to decrease the time complexity, which is $O(bd)$ (where the function O computes the upper bound of the time required to execute all the algorithmic steps, b called branching factor denotes the number of assigns per node and the exponent d denotes the lowest total cost of the optimal path), with the same efficiency we diminish the admissibility criterion by modifying the threshold to (ε+1). Moreover, applying dynamic weighting on heuristic function as implemented in (4).

$h'\quad (x)=[\varepsilon\ .\mathrm{w}(x)+1]h(x),$

(4)

Where $\mathrm{w}(x)$ is the weight function (5).

$\mathrm{w}(x)=\{1-d(x)n, if\ d(x)\leq n0,\ alternatively$

(5)

Where the function $d(x)$ denotes the depth of the search for x nodes, and n indicates the total cost of the optimal path. Hence, the heuristic function $h'\quad (x)$ is optimised following the condition of (6).

$|h'\quad (x)-h(x)|=\mathcal{O}(\log(h'\quad (x)))$

(6)

B. Haversine Formula

By utilising (7) called haversine formula (let $\varphi=dr/$ be the angle between two points on a sphere, where d denotes the distance between the points and r denotes the radius of the sphere) from the mathematical firmament and with the longitudes represented by y and latitudes denoted by x given, we achieve the calculation of the optimal circle distance on Earth's surface by (9). The formula for applying Haversine to the distance calculation can also be represented by either arctangent or spherical low of cosine [46]. According to the roughly spherical shape of the Earth, spherical trigonometry was a prerequisite to accomplishing navigation tasks. Thus, we utilised the haversine formula, which can define the shortest distance between any two points on the surface of a sphere. Determines the long distance of the circle by giving the longitudes and latitudes of the two points [47]. Let the central angle φ between any two points on a sphere defined by (8) where l is the distance between the two points along a great circle of the sphere and r is the radius of the sphere [48].

$hav(\varphi)=\sin 2(\varphi 2)$

(7)

$\varphi=lr$

(8)

$d=2R\sin-1(\sin 2(x2-x12)+\cos(x1)\cos(x2)\sin 2(y2-y12))$

(9)

The haversine formula allows the $hav(\varphi)$ to be calculated directly from the latitude ϑ and longitude λ of the two points as can be seen from (11).

$hav(\varphi)=hav(\vartheta 2-\vartheta 1)+\cos(\vartheta 1)\cos(\vartheta 2)hav(\lambda 2-\lambda 1)$

(10)

Where $\vartheta 1$, $\vartheta 2$ are the latitude of the first and second point, respectively and $\lambda 1$, $\lambda 2$ are the longitudes of the same points.

However, the haversine formula cannot evaluate the actual distance with accuracy ($error$=±0.5%) due to the irregularly ellipsoid shape of the Earth. Therefore, we have developed Vincenty's formulae which consider the ellipticity of the Earth and thus generate more accurate results.

C. Vincenty Formulae

Thaddeus Vincenty, in 1975 presented two iterative methods to solve the direct and inverse problem of geodesics for distances on the ellipsoidal model of the Earth [31]. The methods assume that the figure of the Earth is an oblate spheroid and hence are more accurate than methods that assume a spherical Earth, such as great-circle distance. The direct method calculates the location of a given distance point and azimuth from another point. The inverse method computes the geographical distance and azimuth between two given points [49]. They have been widely used in geodesy because they are accurate within 0.5 mm on the Earth's ellipsoid. Mathematically the geodesic problems are formed as follows:

1. Direct: Given $A(\varphi1,\lambda1)$, $a1$ and s, we would like to calculate $B(\varphi2,\lambda2)$ and $a2$

2. Inverse: Given $A(\varphi1,\lambda1)$ and $B(\varphi2,\lambda2)$, we would like to calculate $a1$, $a2$ and s

For the direct problem, which is to determine the endpoint $B(\varphi2,\lambda2)$ and the corresponding azimuth, $a2$ from a starting point $A(\varphi1,\lambda1)$ with azimuth $a1$. We initially calculate the values given by the equations (11) - (16). The geodesic problem is illustrated in Fig. 1, which depicts the starting point A at latitude $\varphi1$, longitude $\lambda1$ and azimuth $a1$ respectively the point B has latitude $\varphi2$ longitude $\lambda2$ and azimuth $a2$. The ellipsoidal distance between the two endpoints called geodesic is denoted by s.

$U1,2=\tan-1[(1-f)\tan\varphi1,2]$ (11)

$\sigma1=\tan-1(\tan U1\cos a1)$ (12)

$a=\sin-1(\cos U1\sin a1)$ (13)

$u=\sqrt{\ }\cos2x(x2-y2y2)$ (14)

$A=1+u2214\{212+u2[-768+u2(320-175u2)]\}$ (15)

$B=u2210\{28+u2[-27+u2(74-47u2)]\}$ (16)

Afterwards, by (20), which is the initial angular separation between the points, we apply iteration on (18) - (20) until σ to present sufficient accuracy.

$\sigma0=syA$ (17)

$\sigma=\sigma0+\Delta\sigma$ (18)

$2\sigma m=2\sigma1+\sigma$ (19)

$\Delta\sigma=B\sin\sigma\{\cos(2\sigma m)+B4(\cos\sigma[-1+2\cos2(2\sigma m)]-B6\cos(2\sigma m)(-3+4\sin2\sigma)[-3+4\cos2(2\sigma m)])\}$ (20)

Hence, we evaluate the endpoint $(\varphi2,\lambda2)$ and the corresponding azimuth, $a2$.

Concerning the inverse problem, we need to calculate the azimuths a1, a2 and the connecting geodesic s. Initially, given points A(ϕ1,λ1) and B(ϕ2,λ2) we compute, using (11), the reduced latitude on an auxiliary sphere denoted by U1 and U2 and we apply iteration on (21) - (25) until λ converges to the required degree of accuracy (≈0.6mm).

σ=cos−1(sin$U$1sin$U$2+cos$U$1cos$U$2cosλ)⟺sin2σ=(cos$U$2sinλ)2+(cos$U$1sin$U$2+cos$U$2sin$U$1cosλ)2

(21)

a=sin−1(cos$U$1cos$U$2sinλsinσ) (22)

C=f16cos2a[4+f(4−3cos2a)] (23)

σm=12cos−1(cosσcos2a−2sin$U$1sin$U$2cos2a) (24)

λ=λ0+(1−C)$f$sin$a${σ+Csinσ[cos(2σm)+Ccosσ(−1+2cos2(2σm))]} (25)

Where λ0 is the initial value of λ. Hence, after the required iterations, we compute (14) - (16) and (20), from which we can calculate the azimuths a1, a2 and the connecting geodesic s using (26) - (28). For (27) and (28) the comma separates the cartographical coordinates as we calculate the cartographical azimuth, also called grid azimuth. It is commonly used in triangulation and azimuth identification, especially in radar applications.

TABLE I. GLOSSARY OF NOTATIONS Symbol	Quantity
x,y	major and minor semiaxes of the ellipsoid
λ1,λ2	longitude of the points
φ1,φ2	latitude of the points
f=(x−y)x/	flattening of the ellipsoid
φ	geodetic latitude
U1,U2	reduced latitudes on an auxiliary sphere
a1,a2	azimuths of the geodesic
a	equatorial azimuth
λ	the difference in longitude on an auxiliary sphere
s	ellipsoidal distance between the two points
σ	the angular distance between points
σ1	the angular distance between the point

	and the equator
σm	the angular distance between the midpoint and the equator
w	waypoints
O	centre of the obstacle
$\mathcal{O}$	the angle from the connection of the UAV and obstacle centre to the geography north
$\mathcal{D}$	direction angle
ϕ	course angle of the UAV
$RUAV$	radius of UAV
$Robs$	the maximum radius that obstacle can have
l	distance between the two points along a great circle of the sphere
r	radius of the sphere
ω	the angle between the position vector $AO^{\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow}$ and the vector of the UAV’s current location to the next waypoint

s=yA(s−Δσ) (26)

a1=tan−12(cos$U$2sinλ,cos$U$1sin$U$2−cos$U$2sin$U$1cosλ) (27)

12tan$a$2=cos$U$1sinλ,cos$U$1sin$U$2cosλ−cos$U$2sin$U$1 (28)

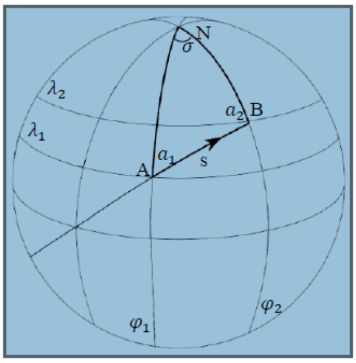


Fig. 1. Illustration of the geodesic problem

Every notation used within equations (11) - (28) is indicated in TABLE I.

PROPOSED SYSTEM ARCHITECTURE

The proposed system architecture utilises drones' current mass production technology to simulate large-scale UAVs that serve near-ground transportation in intelligent urban ecosystems. Hence it will significantly contribute to smart cities' ultimate goal: to improve inhabitants' quality of life as traffic congestion will be resolved or transmitted to the ethers. Based on the last statement, we intend to contribute by studying the problem that arises during the operation of unmanned air traffic systems. Our approach aims to resolve several main tasks: avoidance of no-fly zones, computation of the optimal trajectory, and integration with the UAVs to transmit traffic control signals along the trajectory.

A. Preliminaries

The software of the route calculation module and, more specifically, for the visualisation procedure of the terrain map is required to integrate Google maps, Yandex maps or similar satellite maps with a resolution of about 15 meters per pixel. The program's functionality allows displaying the coordinates of the obstacles in the form of latitude/longitude. Initially, we should set the marks for the beginning and end of the route on a map that displays an area of at least $1km2$. Avoidance of restricted areas mapped in the polygon will be applied on the trajectory of the airmobile vehicle. Hence the optimal route will be calculated and displayed, considering the approach to the no-fly zones no closer than 20 m, which should differ from the shortest possible route by no more than 15 m.

We can process the information transmitted from the airmobile vehicle via the reverse channel through the management system on the ground, including latitude, longitude, altitude, azimuth, and roll speed. Also, we can determine the operator's position in real-time by the GPS/GLONASS receiver signal. Therefore, regular two-way communication between the system management and the airmobile app integrated into the UAV system is a prerequisite for the communication channel. In addition, the transmission speed must be sufficient to control and respond to deviations from the route reliably.

B. System Design

The system is made for GPS-driven drones to levitate around the user-made custom "Red zones" or no-flight zones. Therefore, constructing a path from the drone's current position to the destination is feasible. The path can only be constructed when both the status and the destination point are present—several points consisting of the path located in an invisible grid and connected by straight lines. When the path has been created, the data of the GPS locations of the path points will be sent to the drone using a Wi-Fi chip. Once the drone receives the signal, it obtains the locations of the path points in a stack and follows the path. Each time the drone gets close enough to the point, the point location will be popped. However, due to the inaccuracy of the GPS receiver in the drone, the path will only be followed approximately, and when the drone gets close enough (1-2 meters to the point), the point will be counted as completed, and the drone will fly to the next point.

The GPS signal is received by the Arduino-supporting chip connected to the GY-NEO6MV2 receiver connected to the antenna. The main chip is Node MCU, which has built-in Wi-Fi and allows one to connect to the computer or use multiple chips to create a network. This project is used to transmit data from the stationary chip connected to the PC to another chip inside the drone. The connection is 5V to 5V, RX to TX and Ground to Ground.

C. Algorithmic Implementation

This subsection illustrates a part of the algorithmic procedure used to implement the proposed system. The algorithms include explanations for readability reasons. First, the program calculates the next closest point from the currently searched marker to finish using spherical trigonometry. Then, the cycle repeats until the step distance between the checked marker and the finish is reached. Every time the process counts, one marker is added to the Path_markers array, as presented in Algorithm 1.

The Grid Marker is used to build the grid and calculate the path around the obstacle. Next, we must acquire a physical copy of the object and assign it to that variable by utilising the new operator. Double and float data types represent a Real number; a double data type is more precise than a float. A double variable can provide precision up to 15 to 16 decimal points compared to float accuracy of 6 to 7 decimal digits. Also, double conversion Constant = 0.0111" is used to convert kilometres into Latitude and Longitude later. The difference between Longitudes and Latitudes of extreme points is denoted as $d\varphi$ and $d\lambda$, respectively—the angle ϑ stores the azimuth from start to finish. The float step = 0.0001" defines both the step of the straight line and the distance between neighbouring points on the grid in coordinate units. Store the step values of every new point during the straight path check. Finally, we calculate the new location represented by the Longitude and Latitude of the point on the map. The location class is provided by the library Unfolding_Maps, which is a library to create interactive maps and geo-visualisations in Processing and Java.

Algorithm 1.

```
Input: $\lambda1$,$\lambda2$,$\varphi1$,$\varphi2$
Output: Shortest Path
// This loop (from the next line until the end of the code) is repeated as many times as needed until the distance becomes smaller or equal to the conversion constant
while s>$c1$, $\sigma$>$c2$ do

    //Difference between Longitudes $d\lambda$ and Latitudes $d\varphi$ of extreme points
    $d\lambda$= $\lambda2$−$\lambda1$
    $d\varphi$= $\varphi2$−$\varphi1$

    // $\vartheta$ Stores Azimuth from Start to Finish
    $\vartheta$=tan−1($d\varphi d\lambda$/)

    // If $d\varphi$<0 and $d\lambda$>0 then change the sign of $\vartheta$
    **if** $d\varphi$<0 **and** $d\lambda$>0 **then**
        −$\vartheta$⟵$\vartheta$
    end if

    // if $d\varphi$>0 and $d\lambda$<0 then change the sign of $\vartheta$
    **if** $d\varphi$>0 **and** $d\lambda$<0 **then**
        −$\vartheta$⟵$\vartheta$
    end if

    //Float step = 0.0001”. This defines both the step of the straight line and the distance between neighboring points on the grid (in coordinate units).
    //Store the step values of every new point during the straight path check
    $\varphi s$⟵10−4sin$\vartheta$
```

```
// $fa$: Latitude step converted to float
$fa \longleftarrow \varphi s$
$\lambda s \longleftarrow 10{-}4\cos\vartheta$

// $fo$: Longitude step converted to float
$fo \longleftarrow \lambda s$

//If $d\varphi$ is negative, change the sign of $fa$ (so it will always be positive)
**if** $d\varphi<0$ **then**
    $fa \longleftarrow -fa$
end if

// If $d\lambda$ is negative, change the sign of fo (so it will always be positive)
**if** $d\lambda>0$ **then**
    $fo \longleftarrow -fo$
end if

//Calculate the new location. Location generally used to represent Longitude and Latitude of the point on the map
$\varphi new \longleftarrow \varphi + fa$
$\lambda new \longleftarrow \lambda + fo$
end while
```

Algorithm 2 was applied to avoid obstacles, as the path of the UAV will be predetermined, and collision avoidance is a prerequisite along the trajectory.

Algorithm 2.

```
1: **Set** $W$={$wi$|$i$=1,2,…,$n$}, $\mathcal{O}centre$,$\mathcal{O}radius$

//If the UAV is in the initial phase, it tracks the waypoints

2: for $i$=1 to $n$ do

//Calculate the fluctuation of obstacle's center based on the distance, as a product of speed and time $d$=$vt$

3: $O \longleftarrow O+vt$

// Compute the Euclidean distance from the coordinates of the UAV to obstacle's centre

4: $(\overrightarrow{AO})2=(xA-xo)2+(yA-yo)2$

// We calculate the angle obtained by the line that connects the UAV and obstacle center to the geography north

5: $\mathcal{O}$=tan−1[$(yA-yo)(xA-xo)/$]

//Obtain the direction angle

6: $\mathcal{D} \longleftarrow \mathcal{O}-\phi$

//If the Euclidian distance from line 4 is less or equal to UAV's radius and the sin of the direction angle less than the quotient $RobsRUAV/$ then the UAV pass to the second phase and operates the circular path's maneuver clockwise. Alternatively, it tracks the circular path anti-clockwise.

7: **if** $|\overrightarrow{AO}| \leq RUAV$ **and** sin$\mathcal{D}<RobsRUAV/$ **then**

8: circular maneuver clockwise

9: else
```

```
circular maneuver anti-clockwise
end if

//Compute the angle between the position vector $\overrightarrow{AO}$ and the vector of the UAV's current location to the next waypoint
$\omega \longleftarrow \tan-1[wi+1x-xAwi+1y-yA]-\mathcal{O}$

// If $\omega$>90°, the avoidance of the obstacle is successful.
Alternatively, the UAV continues with the initial phase of its trajectory.
if $\omega$>90° **or** $|\overrightarrow{AO}|\leq RUAV$
    $i \longleftarrow i+1$
end if
end for
```

We initially set the waypoints wi and the prerequisite data to interpret the obstacle, which is the centre and the maximum radius that the obstacle could have denoted as $\mathcal{O}$centre,$\mathcal{O}$radius. $\mathcal{D}$ is the angle of the direction and varies within the interval [−180°,180°]. It is defined as the difference between the angle of the line that links the UAV and the center of the obstacle with the north line denoted by $\mathcal{O}$, and ϕ, the angle of UAV's trajectory. As presented in Fig. 2, where we illustrate the detection of the obstacle and the trajectory that the UAV follow to avoid it, $RUAV$ and $Robs$ are the radius of the UAV and the maximum potential radius of the obstacle, respectively.

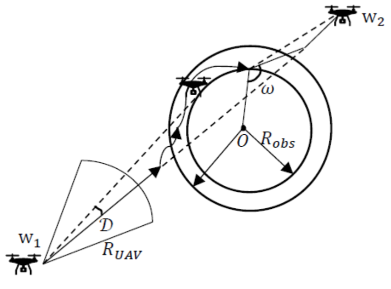


Fig. 2. Obstacle avoidance

Fig. 3 is the visual illustration of the algorithmic technique. C is the current location; D is the destination, and orange and red squares denote the obstacle nodes. The forward search starts from C. Grey points are undiscovered. Firstly, the neighbors of the currently searched points of C are found. They are marked purple. Each of these points is now added to a queue and will appear one by one in the order they were found. In our case, the order is: top, bottom, left, right, upper-left, upper-right, lower-left, lower-right. In addition, those markers now have a state “Frontier”. Finally, the “Undiscovered” neighbours of the second searched point are found and added to the queue.

The point itself became “Discovered” and has a value of 1. This value shows the distance from the starting point and will be used later to find the shortest path. When the first “members” of the queue are processed, the picture will look like this. Notice that the diagonal movements are valued as √2. Following the same algorithm, we expect to get the final image. The algorithm stops when one of the currently checked points appears to be the destination. After that, the path is built following the points with the lowest value. The lowest value shows that the point took the least steps to reach the destination. Searching starts from the Current position (the first point is added). Then the neighbouring points are added to the queue and the presence of Destination point among them is checked. Only the points with state “Undiscovered” are becoming frontiers.

For diagonal movements, the distance is set as static root_2. The X and Y check determines which point is oblique and which is not. If X and Y are changed regarding the starting point, it is diagonal. Also, if the distance to the point is already known and the new length is less, it means that to the checked point, there is a shorter way, and hence a further distance should be applied. This cycle is repeated either until no “Frontiers” are left or until the destination is reached. When the goal is found, the next stage comes. It builds the path according to the lowest distance-price parameter of the next step. The path starts from the destination and goes back to the Current location. The system checks for two parameters: The lowest distance price (it might have several points with the same d-p) and the closeness of the point to the start. It chooses the one closer to the start between two points with the same grid distance. This way, the path is more likely to be shorter. One is added to the final array of Path_markers when every point is checked.

Fig. 3. Visualising parts of the Algorithmic Technique

PERFORMANCE ASSESSMENT

According to the testing depicted in Fig. 4, the test for a common straight path showed an acceptable result. The second test for a standard red zone avoidance also presents an acceptable result, but the path went for two unnecessary points near the current point. Finally, the test for double red zone avoidance in the pic. 3 presents an acceptable result, and the test for a difficult path is also acceptable.

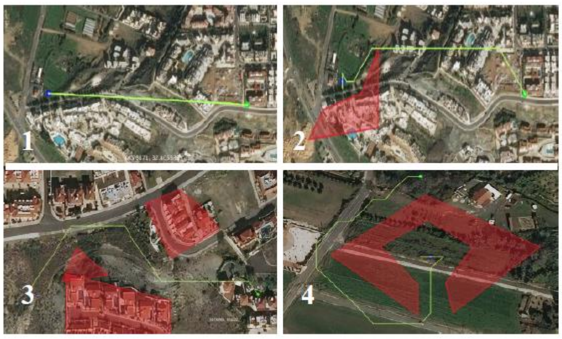


Fig. 4. Testing

TABLE II illustrates in visual presentation the expected and the actual output of the system, including the corresponding efficiency of the program. System validation presents high accuracy for red zone avoidance even if complicated manoeuvres are required. As shown, the accuracy tends to be 100% for straight-line paths, 96% and 95% for more demanding trajectories.

An experimental path planning comparison between A*, improved A* bidirectional A*, lining smoothed A* and lining A* through simulations applied on C++ and MATLAB generated the data presented in TABLE III. The data obtained for computation time, path length, and max and min pitch angle are depicted in Fig. 5, Fig. 6 and Fig. 7. As can be seen from the outcomes, the computation time of the Improved A*, Lining Smoothed A* and Lining A* is the same 6.785 sec. However, between these three modifications of A*, the Improved case presents the longest path due to the trajectory's reconfiguring to avoid the red zones.

TABLE II. SYSTEM VALIDATION

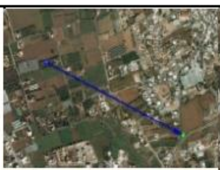
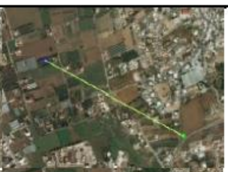
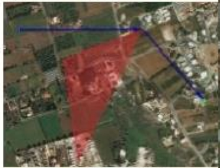
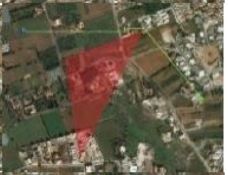
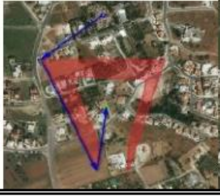
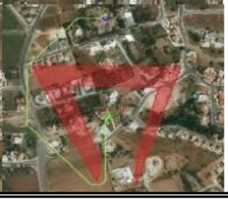
Test description	Expected output	Real output	Accuracy
1.			100%
2.			95%
3.			96%

TABLE III. COMPARING PATH PLANNING

Method	Computation Time (s)	Path Lengths (m)	Max pitch angle (°)	Min pitch angle (°)
A*	29.521	9290	46	-35.15
Improved A*	6.785	10512	30.1	-28.85
Bidirectional A*	23.012	9245	46	-48
Lining Smoothed A*	6.785	8699	30.1	-28.85
Lining A*	6.785	8542	30.1	-28.85

The results indicate that compared with the actual A* algorithm and the bidirectional version, the computation time is reduced by 77.02% and 70.52%, respectively. Also, it is shown that the length of the improved A* algorithm is longer to generate the trajectory of kinematic constraint satisfaction.

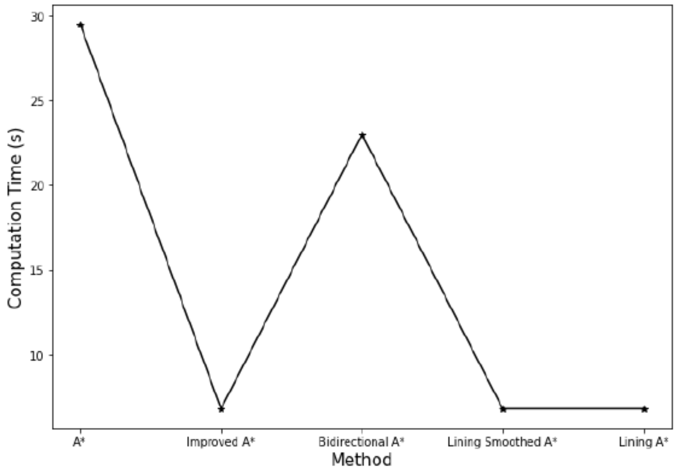


Fig. 5. Computation time

We infer from the max and min pitch angle chart that only the enhanced, lining, and smoothed lining A* algorithms satisfy the pitch angle constraint. Consequently, it indicated better performance by the improved lining and smoothed lining A* algorithms.

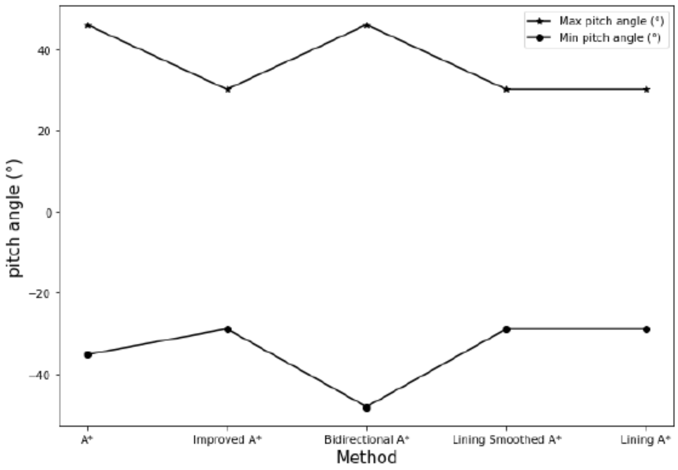


Fig. 6. Pitch angle

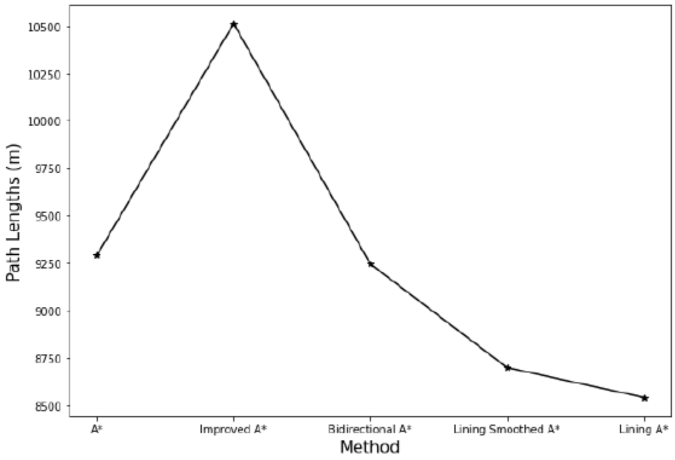


Fig. 7. Path lengths

Relying on the results, we plan to focus our future orientation on non-GPS regions like mountains, deserts, and beaches, for which the network's availability is limited. Therefore, we will thoroughly elaborate on automated guided vehicle communications and networks. This field of research has attracted extensive attention and has broad prospects in the wireless transmission area, especially in areas with limited network access. More specifically, we will study an automated guided unmanned aerial vehicle communication scenario in which a group of automated guided vehicles will form a virtual antenna array and communicate with different unmanned aerial vehicles. These will be based on distributed collaborative beamforming.

Simulation Paradigm

We have implemented a scenario in MATLAB Simulink using a moving target for landing. As seen in Fig. 8, we have a UAV levitate on the y-axis 500m above the ground with a speed u=267.3m/s. Directly opposite we have a Jammer at 0.89km and at the ground level the target for landing is moving away from it through a high-rise buildings area. The moment that we are investigating is 0.75km away from the UAV. We consider the area 0.16m2 from the origin as the tower block with a Clutter Gamma −26 dB, thus, the UAV cannot land within this area. The jamming signal is correlated with a specific direction. Although, due to the white noise nature of the jammer, the received jamming signal dominates the entire Doppler band.

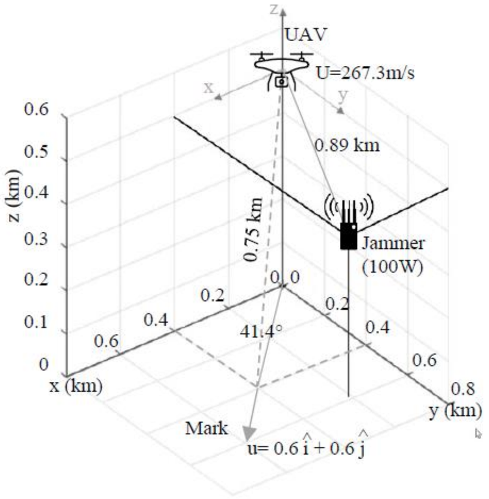


Fig. 8. Paradigm with a moving target for landing

Different stages of simulation generate the results that we present below. Initially, Fig. 9 displays how the signal obtained at the antenna array is predominated by the clutter return. Since the UAV is levitating approximately 500m above ground level the clutter returns from the surface and starts at that high.

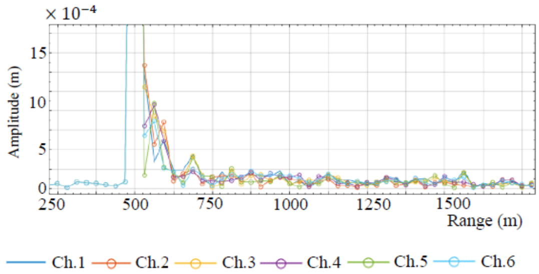


Fig. 9. Different stages of the simulation

We evaluated the proposed algorithmic technique through simulations for a rectilinear moving target. The target was operating discrete speed cases. In addition, we implemented the process for a circular trajectory moving target with a variety of minimum turning radii. In each scenario, the UAV was initialised at an altitude of 500m. The main objective was the avoidance of obstacles during the procedure of approaching the moving target for landing. Thus, the proposed algorithmic technique was deployed to configure the relative position between the UAV and the target considering the real-time range and the bearing angle. The centre of the new circular path could also be determined if the predicted range between the UAV and the moving target couldn't meet the requirements. The second proposed algorithm was applied for planning the transition path between circular paths that constrain the turning radius of the UAV. Hence, we generated the waypoints that meet the flight ability. We also formulated an observation angle as an evaluation index. Therefore, this series of simulations andevaluation index comparisons verify the effectiveness of the proposed algorithms.

TABLE IV presents the statistical results obtained from Fig. 10, which displays the return at the output of the Adaptive Displaced Phase Center Array (ADPCA) pulse canceller for a uniform linear array, clearly showing the target's range at approximately 500m. Also, we have been filtered out the simplest form of jamming that is a barrage jammer. It is an intense, continuous white noise directed toward the UAV's radar receiver.

TABLE IV. STATISTICAL RESULTS FROM Fig. **10**

Ch.1	Max	Min	Standard Deviation	Mean
x	5×10^2	2.7×10^2		
y	1×10^{-6}	9.1×10^{-9}	1.1×10^{-8}	9×10^{-8}

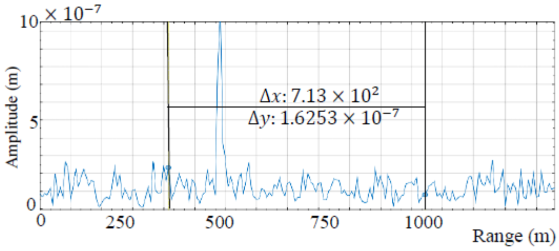


Fig. 10. Return at the output of the Adaptive Displaced Phase Center Array (ADPCA) pulse canceller.

Depicting the clutter as a function of angle and Doppler shift shows that the clutter return looks like a diagonal line in the angle-Doppler environment, called clutter ridge. Findings show that the received jammer signal is a white noise spread over the entire Doppler range at 60o.

In addition, we generated a deep null along the clutter ridge as well as in the direction of the jammer. It is based on the Adaptive displaced phase centre array canceller weights. We modelled through MATLAB Simulink a monostatic radar with a moving target and a static jammer that transmits jamming signals through free space to the radar. A six-element uniform linear antenna array with back-aperture elements then receives the reflected pulse from the target and interferes with the jammer. The output of a clutter simulator is also added to the received signal before it is processed. After adding noise, the signal is stored in a data repository. The repository is processed by the adaptive shifted centre phase block canceller to the target's estimated range, azimuth angle, and doppler shift. Therefore, the canceller is scanning multiple ranges, azimuth angles, and doppler shifts as the target's velocity and position are unknown.

CONCLUSION & DISCUSSION

This research has presented a comprehensive study on the upcoming near-ground air mobility in Smart Cities. We have focused on developing a system that could efficiently enable the correction and determination of the trajectory for UAVs. It was implemented by modifying the A* algorithmic technique and elaborating Haversine and Vincenty's formulas. Consequently, tests have shown that the program works effectively even in difficult situations like the last tests. Most of the functionality required has been introduced into the system and functions as intended by the technical specifications. However, the system could be improved by the ability to change the step of the grid using the GUI. Also, instead of the drone, we could create a drone simulation, which could be displayed on the interface and move in real-time. Also, the positions of the red zones could be saved into the file and loaded back into the system. The simulation paradigm proves that the proposed algorithmic technique functions efficiently. Elaborating on the scenario of landing on a moving target and avoiding obstacles has added more significance to our research approach. Results and outcomes from the testing cases demonstrate that the current method outperforms current state-of-the-art methodologies for fully autonomous UAVs. By accurately determining their trajectory, their effectiveness will be increased in realising their diverse operations and capabilities in the Smart Cities ecosystem.

The embodiment of UAVs within complex dynamic environments such as Smart Cities with which they will interact in real-time for various operations conceals unknown obstacles. Current computing technological systems cannot define the critical path by identifying barriers. Therefore, the above vehicles can also self-locate the obstacles and determine the optimal trajectory. Hence, it is necessary to allow essential operations to be performed autonomously on UAVs by activating IoT communication for data acquisition. The required data can be obtained from sensors or Global Positioning Systems (GPS), which will be vital for achieving efficiency in crucial operations. The rise and the imminent autonomy control levels are illustrated in Fig. 11. UAVs can perform vertical take-off and landing, trajectory pre-planning, real-time data dissemination etc. However, it is paramount to enable their ability to execute AI-based operations to detect the environment and computing in IoT network systems in the optimal trajectory.

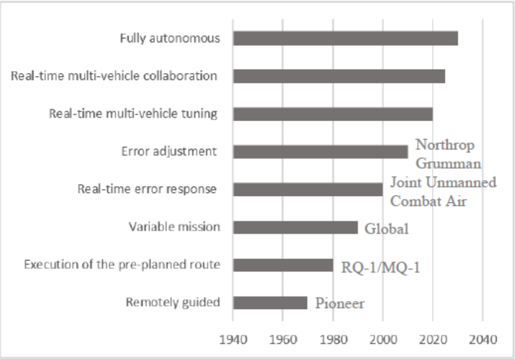


Fig. 11. The rise and the upcoming autonomous control systems

FUTURE ORIENTATION

We are planning to formulate an automated guided unmanned aerial vehicle communication with a multi-objective optimisation problem to maximise the total transmission rate simultaneously, minimise the total repositioning time of automated guided vehicles and minimise the total motion energy consumptions by optimising the locations, current innervation weights and moving speeds, as well as the sequence for communicating with different vehicles. Based on the mathematical firmament, we will elaborate on the multi-objective antlion optimisation algorithm with Chebyshev chaos-opposition-based learning solution initialisation and hybrid solution update method.

REFERENCES

[1] P.-V. Mekikis, A. Antonopoulos, E. Kartsakli, L. Alonso and C. Verikoukis, “Communication recovery with emergency aerial networks,” IEEE Transactions on Consumer Electronics, vol. 63, no. 3, pp. 291 - 299, Aug. 2017.

[2] S. Khastgir, S. Brewerton, J. Thomas and P. Jennings, “Systems Approach to Creating Test Scenarios for Automated,” Reliability Engineering and System Safety, 16 Mar. 2021.

[3] Y. Rasekhipour, A. Khajepour, S.-K. Chen and a. B. Litkouhi, “A Potential Field-Based Model Predictive Path-Planning Controller for Autonomous Road Vehicles,” IEEE Transactions on Intelligent Transportation Systems, 05 May 2017.

[4] R. Bogue, “Robots to aid the disabled and the elderly,” Industrial Robot, 14 Oct. 2013.

[5] A. Mardani and S. Ebrahimi, “A Novel Multimode Mobile Robot with Adaptable Wheel Geometry for Maneuverability Improvement,” International Journal of Robotics, Theory and Applications, vol. 4, no. 4, pp. 1 - 15, 2016.

[6] Q. Ju and Y. Zhang, “Clustered Data Collection for Internet of Batteryless Things,” IEEE Internet of Things Journal, vol. 4, no. 6, Dec. 2017.

[7] C. H. Kim and B. K. Kim, “Minimum-Energy Translational Trajectory Generation for Differential-Driven Wheeled Mobile Robots,” Journal of Intelligent and Robotic Systems, vol. 49, no. 4, p. 367–383, 2007.

[8] Q. Wu, J. Xu, Y. Zeng, D. W. K. Ng, N. Al-Dhahir, R. Schober and L. Swindlehurst, “A Comprehensive Overview on 5G-and-Beyond Networks With UAVs: From Communications to Sensing and Intelligence,” IEEE Journal on Selected Areas in Communications, vol. 39, no. 10, pp. 2912-2945, Oct. 2021.

[9] Y. Zeng, Q. Wu and R. Zhang, “Accessing From the Sky: A Tutorial on UAV Communications for 5G and Beyond,” Proceedings of the IEEE, vol. 107, no. 12, pp. 2327-2375, 2019.

[10] G. Khayat, C. X. Mavromoustakis, G. Mastorakis, J. M. Batalla and E. Pallis, “Swarm UAV Network Constraints in Damaged Infrastructures,” in IEEE International Conference on Communications, Seoul, Korea, Republic of, 2022.

[11] G. Khayat, C. X. Mavromoustakis, G. Mastorakis, J. M. Batalla and E. Pallis, “Weighted Cluster Routing Protocol with Redundant Cluster Head for Damaged WSN,” in IEEE 7th International Energy Conference, Riga, Latvia, 2022.

[12] X. Xiong, C. Sun, W. Ni and X. Wang, “Three-Dimensional Trajectory Design for Unmanned Aerial Vehicle-Based Secure and Energy-Efficient Data Collection,” IEEE Transactions on Vehicular Technology, pp. 1-15, 02 Sep. 2022.

[13] W. Wen, K. Luo, L. Liu, Y. Zhang and Y. Jia, “Joint Trajectory and Pick-up Design for UAV-assisted Item Delivery under No-Fly Zone Constraints,” IEEE Transactions on Vehicular Technology, pp. 1-6, 04 Oct. 2022.

[14] I. Bisio, H. Haleem, C. Garibotto, F. Lavagetto and A. Sciarrone, “Performance Evaluation and Analysis of Drone-Based Vehicle Detection Techniques From Deep Learning Perspective,” IEEE Internet of Things Journal, vol. 09, no. 13, pp. 10920 - 10935, 01 Jul. 2022.

[15] E. W. Dijkstra, “A note on two problems in connexion with graphs,” Numerische Mathematik, p. 269–271, 1959.

[16] Y. Li, Y. Wu, X. Su and J. Song, “Path Planning for Aircraft Fleet Launching on the Flight Deck of Carriers,” Mathematics, vol. 6, no. 10, 25 Sep. 2018.

[17] P. Yao and S. Zhao, “Three-Dimensional Path Planning for AUV Based on Interfered Fluid Dynamical System Under Ocean Current,” IEEE Access, vol. 6, pp. 42904-42916, 2018.

[18] M. Dorigo and L. M. Gambardella, “Ant colony system: a cooperative learning approach to the traveling salesman problem,” IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, pp. 53-66, 1997.

[19] G. Pan and M.-S. Alouini, “Flying Car Transportation System: Advances, Techniques, and Challenges,” IEEE Access, vol. 9, pp. 24586-24603, 03 Feb. 2021.

[20] C. Luo, J. Nightingale, E. Asemota and C. Grecos, “A UAV-Cloud System for Disaster Sensing Applications,” in IEEE 81st Vehicular Technology Conference, 2015.

[21] M. Faisal, M. Salleh, C. Wanchao, Z. Wang, S. Huang, D. Y. Tan, T. Huang and K. H. Low, “Preliminary Concept of Adaptive Urban Airspace Management for Unmanned Aircraft Operations,” in AIAA Information Systems-AIAA Infotech, Aerospace, 2018.

[22] M. Hatfield, C. Cahill, P. Webley, J. Garron and R. Beltran, “Integration of Unmanned Aircraft Systems into the National Airspace System-Efforts by the University of Alaska to Support the FAA/NASA UAS Traffic Management Program,” Remote Sensing, vol. 12, 2020.

[23] J. Lieb and A. Volkert, “Unmanned Aircraft Systems Traffic Management: A comparsion on the FAA UTM and the European CORUS ConOps based on U-space,” in IEEE 39th Digital Avionics Systems Conference, 2020.

[24] A. S. Shahriar, H. Kevin, F. Grigorios, E. Ugur, B. Irina, S. Stephen and A. Panagiotis, “The Flying Car—Challenges and Strategies Toward Future Adoption,” Frontiers in Built Environment, vol. 6, 2020.

[25] M. N. P. a. G. Sarné, “Reinventing Mobility Paradigms: Flying Car Scenarios and Challenges for Urban Mobility,” Sustainability, vol. 12, no. 9, 28 Apr. 2020.

[26] A. Mofolasayo, “Potential Policy Issues with Flying Car Technology,” Transportation Research Procedia, vol. 48, 2020.

[27] X. Chen, W. Cao, Q. Zhang, S. Hu and J. Zhang, “Artificial Intelligence-Aided Model Predictive Control for a Grid-Tied Wind-Hydrogen-Fuel Cell System,” IEEE Access, vol. 8, pp. 92418-92430, 2020.

[28] A. Arshad, A. J. Kallungal and A. A. A. E. Elmenshawy, “Stability Analysis for a Concept Design of Vertical Take-off and Landing (VTOL) Unmanned Aerial Vehicle (UAV),” in 2021 International Conference on Military Technologies (ICMT), 2021.

[29] A. Andreou, C. Mavromoustakis, G. Mastorakis, E. Pallis, N. Magaia and E. Markakis, “Intelligently Reduce Transportation’s Energy Consumption,” in Intelligent Technologies for Internet of Vehicles. Internet of Things (Technology, Communications and Computing), Springer, Cham, 2021.

[30] L. Euler, “Elements of spheroidal trigonometry taken from the method of maxima and minima,” History of the Royal Academy of Sciences and Belles-Lettres in Berlin, vol. 9, p. 258–293.

[31] T. Vincenty, “Direct and Inverse Solutions of Geodesics on the Ellipsoid with application of nested equations,” Survey Review, vol. 23, no. 176, p. 88–93, Apr. 1975.

[32] A.-M. Legendre, "Analyse des triangles tracés sur la surface d’un sphéroïde," Mém de l’Inst Nat de France, vol. 1, pp. 130-161, 1806.

[33] B. Oriani, " Elementi di trigonometria sferoidica Pt 1, 2, 3," Mem dell’Ist Naz Ital, Vols. 1, 2, 2, no. 1, 1, 2, pp. 118–198, 1–58, 1–58, 1806, 1808, 1810.

[34] F. W. Bessel, “The calculation of longitude and latitude from geodesic measurements,” Astron Nachr, vol. 4, no. 86, p. 241–254, 1825.

[35] F. R. Helmert, Die Mathematischen und Physikalischen Theorieen der Höheren Geodäsie, vol. 1, Leipzig, 1884.

[36] P. Nikolaidis and A. Poullikkas, “Cost metrics of electrical energy storage technologies in potential power system operations,” Sustainable Energy Technologies and Assessments, vol. 25, pp. 43-59, 2018.

[37] X. Yang, J. Chen, P. Su, X. Lei, J. Lei, Y. Xie and a. Z. Z. Jianping Yuan, “Detection of Defects in Film-Coated Metals and Non-Metallic Materials Based on Spoof Surface Plasmon Polaritons,” IEEE Sensors Journal, vol. 19, no. 24, pp. 11891 - 11899, 15 Dec. 2019.

[38] B. Kiumarsi, K. G. Vamvoudakis, H. Modares and a. F. L. Lewis, “Optimal and Autonomous Control Using Reinforcement Learning: A Survey,” IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 6, pp. 2042 - 2062, 07 Dec. 2017.

[39] H. Kim, A. Perry and P. and Ansell, “A Review of Distributed Electric Propulsion Concepts for Air Vehicle Technology,” in AIAA/IEEE Electric Aircraft Technologies Symposium (EATS), 2018.

[40] M. Dumke, G. Trigo, M. Sagliano and P. S. a. S. T. Theil, “Design, development, and flight testing of the vertical take-off and landing GNC testbed EAGLE,” CEAS Space Journal volume, p. Theil , 2020.

[41] A. Qazi, F. Hussain, N. Rahim, G. Hardaker, D. Alghazzawi, K. Shaban and K. Haruna, “Towards Sustainable Energy: A Systematic Review of Renewable Energy Sources, Technologies, and Public Opinions,” IEEE Access, vol. 7, pp. 63837-63851, 2019.

[42] H. Baomar and P. J. Bentley, “Autonomous navigation and landing of large jets using Artificial Neural Networks and learning by imitation,” in 2017 IEEE Symposium Series on Computational Intelligence (SSCI), 2017.

[43] P. V. Mekikis, A. Antonopoulos, E. Kartsakli, L. Alonso and C. Verikoukis, “Communication recovery with emergency aerial networks,” IEEE Transactions on Consumer Electronics, vol. 63, no. 3, pp. 291 - 299, Aug. 2017.

[44] E. Dijkstra, “A note on two problems in connexion with graphs.,” Numerische Mathematik, vol. 1, p. 269–271, Dec. 1959.

[45] P. Hart, N. Nilsson and B. Raphael, “A Formal Basis for the Heuristic Determination of Minimum Cost Paths,” IEEE Transactions on Systems Science and Cybernetics, vol. 4, no. 2, pp. 100 - 107, July 1968.

[46] D. Ikasari, Widiastuti and R. Andika, “Determine the Shortest Path Problem Using Haversine Algorithm, A Case Study of SMA Zoning in Depok,” in 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 2021.

[47] V. Hassija, V. Saxena, V. Chamola and F. R. Yu, “A Parking Slot Allocation Framework Based on Virtual Voting and Adaptive Pricing Algorithm,” IEEE Transactions on Vehicular Technology, vol. 69, no. 06, pp. 5945 - 5957, 09 Mar. 2020.

[48] X. Qu, E. Liu, R. Wang and H. Ma, “Complex Network Analysis of VANET Topology With Realistic Vehicular Traces,” IEEE Transactions on Vehicular Technology, vol. 69, no. 04, pp. 4426 - 4438, 28 Feb. 2020.

[49] S.-R. Yang, Y.-J. Su, Y.-Y. Chang and H.-N. Hung, “Short-Term Traffic Prediction for Edge Computing-Enhanced Autonomous and Connected Cars,” IEEE Transactions on Vehicular Technology, vol. 68, no. 4, pp. 3140 - 3153, 13 Feb. 2019.