UAV GNSS Position Corrections based on IoT LoRaWAN^{TM} Communication Protocol

António Raimundo
ISCTE-IUL
Av. das Forças Armadas
Lisbon, Portugal
Instituto de Telecomunicações
Av. Rovisco Pais, 1, Lisbon, Portugal
Email: aslro@iscte-iul.pt

Daniel Fernandes
ISCTE-IUL
Av. das Forças Armadas
Lisbon, Portugal
Instituto de Telecomunicações
Av. Rovisco Pais, 1, Lisbon, Portugal
Email: dfsfs@iscte-iul.pt

David Gomes
ISCTE-IUL
Av. das Forças Armadas
Lisbon, Portugal
Instituto de Telecomunicações
Av. Rovisco Pais, 1, Lisbon, Portugal
Email: dgsda1@iscte.pt

Octavian Postolache
ISCTE-IUL
Av. das Forças Armadas
Lisbon, Portugal
Instituto de Telecomunicações
Av. Rovisco Pais, 1, Lisbon, Portugal

Pedro Sebastião
ISCTE-IUL
Av. das Forças Armadas
Lisbon, Portugal
Instituto de Telecomunicações
Av. Rovisco Pais, 1, Lisbon, Portugal

Francisco Cercas
ISCTE-IUL
Av. das Forças Armadas
Lisbon, Portugal
Instituto de Telecomunicações
Av. Rovisco Pais, 1, Lisbon, Portugal

***Abstract*—With the increase of devices connected to the Internet, currently known as Internet of Things (IoT), it is important to use algorithms in data transmissions to achieve optimal results accordingly to specific use case applications.**

This paper research main goal is to gather Global Navigation Satellite System (GNSS) positions at maximum rate possible, process them in a Post-processed Kinematics (PPK) environment, in order to perform GNSS corrections when compared to real earth GNSS coordinates. This method translates as micro-adjustments of the Unmanned Aerial Vehicle (UAV) traveled path, achieving centimeter-level accuracy GNSS actual positions. These GNSS positions are collected by the UAV GNSS receiver, and then sent to a gateway by using LoRaWAN^{TM} communications protocol.

***Index Terms*—IoT, LoRaWAN^{TM}, UAV, Communications Protocol, PPK**

I. Introduction

There is currently an increase in devices that are connected to the Internet. Among these devices are various types of sensors such as temperature and humidity sensors that are integrated in the smart cities and smart home platforms. Nowadays the Internet of Things (IoT) extends Internet connectivity beyond traditional devices like desktop and laptop computers, smartphones and tablets to a diverse range of devices and everyday things that are using sensors and embedded technology to communicate and interact with the external environment, all via the Internet this technology being also adopted in this work.

One of the important aspects to take into account in IoT is how data is transmitted from the sensors / devices to other components of the IoT ecosystem such as IoT edge or IoT cloud. There are several communication protocols already established such as Wireless Fidelity (Wi-Fi), Bluetooth and Long-Range Wide Area Network (LoRaWAN^{TM}). However, all protocols have its own advantages and disadvantages, depending on the use cases.

This work aims to track the movement of a Unmanned Aerial Vehicle (UAV, also known as drone) using a Global Navigation Satellite System (GNSS) sensor to perform navigational readings of UAV's current earth position. These readings are transmitted through by LoRaWAN^{TM} protocol to test the efficiency and reliability of the LoRaWAN^{TM} gateway, in order to perform post-processed position corrections, known as Post-Processed Kinematic (PPK). PPK systems are becoming the most precise systems to perform precision map topography of a known area [1]. Applications such as agriculture surveys and mapping can deliver centimeter-level accuracy maps. A company named ArnsTronic was capable of drawing planting lines of a 150 ha field with a 4 cm accuracy, saving time, labor and maintenance costs [2].

In this paper, commonly used IoT-ready devices communication protocols will be discussed, followed by its advantages and disadvantages. The implemented system follows a process workflow, represented by an architecture schematic of the communication protocol and data retrieving. The test case and various scenarios, followed by its experimental results are also represented. Finally, the main conclusions and future work are also presented.

II. Communication Protocols for IoT

The main characteristics of the communication protocols that can be used in IoT are presented in Table I [3], [4].

Wi-Fi (IEEE 802.11) [5], is a communication protocol that allows devices to interconnect on a wireless local area network

TABLE I
MAIN CHARACTERISTICS OF THE COMMUNICATION PROTOCOLS

Characteristic	Wi-Fi	Bluetooth	LoRaWAN™
Data Rate [kbps]	11×10^3	1×10^3	27
Frequency [GHz]	2,4 or 5	2,4	0,868
Range [m]	1-100	10-100	20000
Power Consumption [mA]	100-350	1-10	1-10
Security	WPA/WPA2	128 bit	128 bit

(WLAN). The main advantages of this protocol is the data rate and the main disadvantage is the energy consumption [6].

Bluetooth is another communication protocol vastly used on today's micro devices, and the main reason of its appearance was to replace the cables that connected the devices to the computers namely printers, keyboard and others [6]. One of the most important advantage of Bluetooth is the lower average power consumption, when compared with Wi-Fi. Power efficiency is a common problem in the development of small and mid-sized applications where power draw comes from battery packs or solar energy cells.

One of the disadvantages of Wi-Fi and Bluetooth protocols relates to the maximum communication range. A solution to overcome this problem is the use of Long-Range Wide-Area Network (LoRaWAN™).

LoRaWAN™ allows high network capacity and low power consumption, when compared with other technologies. It has a low cost implementation, making this an important factor in choosing the communication protocol to be implemented in IoT applications [7]. According to [8] end-points devices are defined according to the classes:

- Class A: This devices use bi-directional communication, where each end-device has an uplink transmission followed by 2 downlink receive windows;
- Class B: This devices require a time-synchronized beacon from the gateway in order to enable an extra receive windows at scheduled time;
- Class C: This devices have the highest energy consumption, since, this devices have continually open receive windows, only closed when transmitting. This feature allows devices of this class to have less latency.

For this paper, the LoRaWAN™ communication protocol was used mainly because its advantages over other communication technologies. The most important aspect to consider in PPK systems is the maximum communication range. Longer ranges allows longer UAV flights, which also allows a higher coverage area.

III. SYSTEM ARCHITECTURE

This paper main goal (proposed solution) is to perform UAV post-processing position corrections, by monitoring its trajectory on a known path. The UAV current position readings are sent via LoRaWAN™ communication protocol to a established base station near the testing area, which also has a GNSS receiver. The position error comparison between UAV GNSS sensor and base station GNSS sensor are used to perform the needed corrections. These corrections are made by combining both raw GNSS data of the onboard GNSS receiver module and the base station GNSS receiver, explaining an actual PPK system. To implement the proposed solution, there was a need to install a onboard LoRaWAN™ full-duplex communication platform (known as companion hardware) on a small quadcopter. The UAV system is powered by ArduPilot (ArduCopter) firmware, which is an open-source flight controller software and the Pixhawk flight controller unit hardware. The LoRaWAN™ module used was an Arduino-based Microcontroller Unit (MCU) known as SmartEverything Lion. This MCU's specifications are as described in the list below [9]:

- Microchip CPU Cortex M0+ USB Host (ATSAMD21J18A-MU)
- Microchip Crypto-Authentication (ATSHA204A-MAHDA-T)
- Dynaflex 868 MHz Antenna (915/2)
- Linear Power management (LTC3526LEDC-LTC4413EDD-LTC1844ES5)
- Microchip LoRa Module (RN2483)
- Telit GNSS receiver with Embedded Antenna (SE868-A)
- Microchip Bluetooth Low Energy (RN48730)

The RN2483 LoRaWAN™ component is class A LoRa system, which allows full-duplex communication between LoRaWAN™ devices. The system architecture developed in this paper follows the design depicted on Figure 1.

Fig. 1. Architecture design.

The communication is established between the LoRaWAN™ module and the LoRaWAN™ gateway installed on the established base station.

This module will gather GNSS coordinates over time, creating a long string of comma-separated latitude and longitude double values. This string is encrypted by an Application Session Key (AppSKey), and sent to the base station. The AppSKey is defined and stored on the LoRaWAN™ module firmware. Other values such as End Device Identifier (DevEUI) are also defined and stored on the module [10].

The LoRaWAN™ gateway is working on a redirect-only mode, which means that all LoRaWAN™ messages received from this device are forwarded to an external server via HTTP POST request (REST API), and using JSON-formatted data type. The data received on the REST API are processed by decrypting the message payload, using the shared AppSKey, resulting on getting the original plain text message sent by the LoRaWAN™ module. Finally, the string is converted to a human-readable list of latitude and longitude values.

IV. TESTS AND RESULTS

In order to test the proposed solution, there was a need to choose an open area where an UAV could fly at optimal conditions.

For the various test case scenarios, a parking zone was used, which is an object-free zone at a certain altitude. This parking area is located near ISCTE University Institute of Lisbon, Portugal. Days with good weather conditions were also chosen to achieve the best environmental conditions possible.

A. Tests

On every test case scenario, the UAV started to gather GNSS positions immediately after satellite locks (GNSS fix or 3D Lock). The trajectory chosen for the test was a rectangle-like trajectory aligned with the inner edges of the parking area. ArduCopter firmware allows the UAV to make autonomous flights by choosing a number of GNSS coordinates. These points, properly called *waypoints*, are geographical points (latitude, longitude, altitude). The table below represents each waypoint value, making a path from Point A to Point D, crossing intermediary B and C points. Altitude was fixed to maintain UAV stability and to avoid collision with objects.

TABLE II
GNSS THEORETICAL POINTS

	Latitude	Longitude	Altitude [m]
Point A	38,74970	-9,153300	50
Point B	38,75101	-9,153975	50
Point C	38,75079	-9,155167	50
Point D	38,74949	-9,154447	50

Three different scenarios were performed in order to test LoRaWAN^{TM} implemented system capability to send and receive data in real-time.

This test case uses two distinct variables, such as the sampling frequency and the GNSS data type. Table III refers to the different values defined for each scenario.

TABLE III
TEST SCENARIOS

	Scenario 1	Scenario 2	Scenario 3
Sampling Frequency [Hz]	0,2	0,2	0,5
GNSS Data Type	RAW	Δ_{n-1}	Δ_{n-1}
Position Points	96	96	180

The sampling frequency consists on the number of GNSS position readings per second.

The GNSS data type variable refers to how the data string was built. On scenarios 2 and 3, the data values was generated by calculating the difference of the actual GNSS position and the previous GNSS position collected.

The position points are the total number of GNSS position readings at test case scenario ending.

There were some trial and error tests in which the coordinates were sent as they were returned by GNSS receiver. However, when sending the string with all collected coordinates accordingly to the sampling frequency, which was initially a much higher value, the base station LoRaWAN^{TM} gateway refused the connection. The GNSS receiver module allowed GNSS position readings frequency up to 5 Hz. This allows the system to send precise data over time to the base station. Although, base station only allowed a single transmission per minute, per device. All GNSS positions collected at 5 Hz was appended to a single string, in order to be sent every minute. This method was immediately not possible to use because the string length (in bytes) exceeded the maximum payload allowed per transmission, which was approximately 248 bytes. This restriction affected this paper test cases in general, because the initial goal of this paper was to gather GNSS readings at maximum rate in order to get a more accurate trajectory path before PPK application.

On scenario 1, the GNSS data string was built with GNSS coordinates gathered with only 3 significant decimal digits. The need of satisfying LoRaWAN^{TM} gateway limitations, the use of this transmission data model was the only possible method that made a successful transmission to the base station without being rejected. So the solution to this test scenario was to only send GNSS coordinates with 3 significant decimal digits.

On scenario 2, the value of sampling frequency was equal to scenario 1. The main difference between scenario 2 and scenario 1 was the GNSS data type. On scenario 1, GNSS readings were sent in RAW position values. On scenario 2, the GNSS data that was sent to the LoRaWAN^{TM} was generated by calculating the difference between the current GNSS position and the previous GNSS position. This method reduced drastically the payload length. This method also allowed the message sent to contain more information, thus more GNSS position readings, originating test case scenario 3.

On scenario 3, GNSS positions were collected on a higher sampling frequency. This scenario was the best case scenario, because this method allowed a more accurate path trajectory, due to having two times more GNSS positions collected when compared to scenario 2.

B. Results

The results of the scenarios explained previously are present in the Figures 2, 3 and 4.

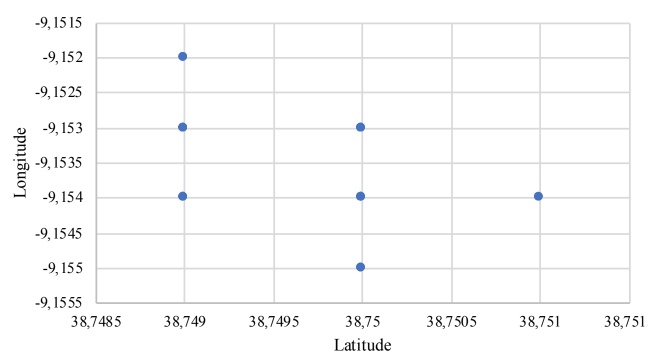

Fig. 2. GNSS positions collected on scenario 1.

On scenario 1, GNSS position coordinates were collected with only 3 significant decimal digits, due to gateway limitations as stated before, resulting on very low trajectory accuracy. During the path realization, the fluctuations at latitude and longitude did not occur on the first significant decimal digits, due to the path nature and earth's coordinating system, resulting on representing only the points received at each transmission iteration, invalidating the trajectory path visualization.

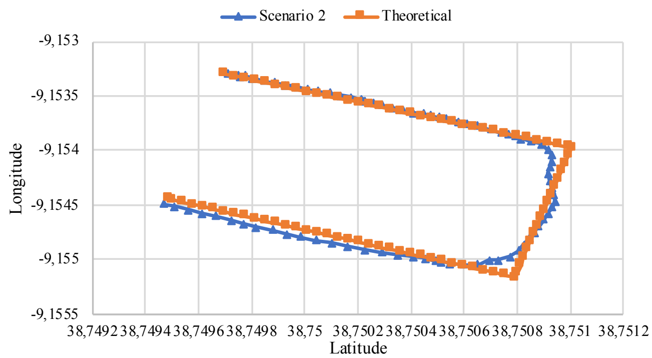

Fig. 3. GNSS positions collected on scenario 2.

On scenario 2 depicted by Figure 3 there are two curves. The blue curve represents the experimental GNSS points collected. The orange curve is the theoretical path which corresponds to the real earth path traveled autonomously by the UAV. Only the differences between the current GNSS position coordinates and the previous one were sent through LoRaWAN^{TM}. It is possible to verify that the points obtained represent the trajectory performed. This scenario represents a great improvement when compared to scenario 1, due to the optimization of the transmitted data model.

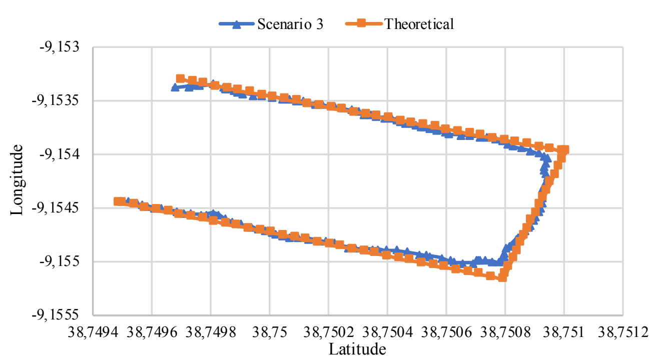

Fig. 4. GNSS positions collected on scenario 3.

On scenario 3, when the sampling frequency is increased, the number of points measured along the path also increases, thus translating into a greater precision in the representation of the performed path. It is possible to verify in the Figure 4 that the density of points increased, when compared to scenario 2 results (depicted on Figure 3) and the obtained path represents more fluctuations.

Through the tool available in [11], Figure 5 shows trajectory paths on different test scenarios, overlaid on a Google Maps map. Theoretical path is represented by the red lines. Scenario

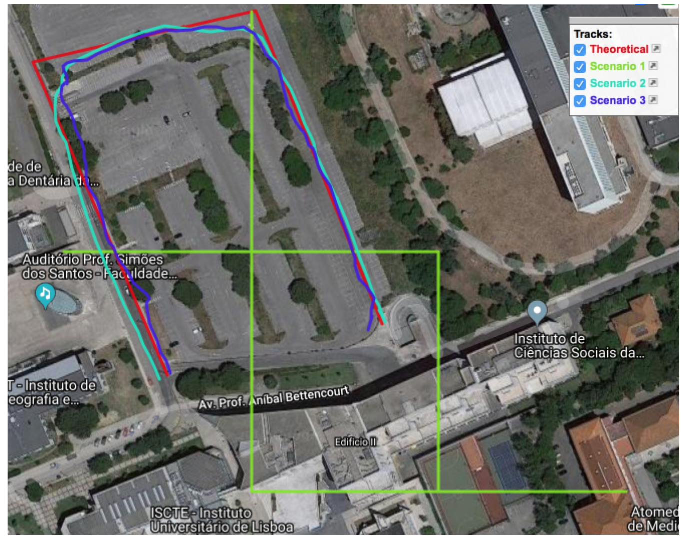

Fig. 5. Test case scenarios trajectory comparison map.

1 is represented by the green lines. These lines does not represent an actual path, due to the lack of GNSS coordinates precision gathered over time. The lines between the points are just representations of possible paths taken. Scenario 2 and scenario 3 are represented by teal and blue paths, respectively. These scenarios are similar to the theoretical path, but not as precise. There was a need to measure the error between the two scenarios and the theoretical values in order to evaluate which one was the most accurate. Some intermediary points in theoretical path were generated in order to calculate the distance error between its relative points in each scenario. A total of 25 points were chosen on both scenarios.

The distance between two geographic points is represented in Equation 1 [12]:

$$d[\mathrm{m}] = \arccos(A + B) \times R \tag{1}$$

where

$$A = \sin(T_{Lat}) \times \sin(E_{Lat}),$$

$$B = \cos(T_{Lat}) \times \cos(E_{Lat}) \times \cos(E_{Lon} - T_{Lon})$$

and $R = 6371000$ m, which represents earth radius; T and E represents Theoretical and Experimental latitude / longitude values, when applicable.

In Figure 6, for both scenarios, each point has its own distance error values, when compared with theoretical ones. Two spikes on points 8 and 15 represent the highest distance error values on both scenarios, fixing its maximum at approximately 30 m distance error. The minimum values were obtained at trajectory starting points.

Table IV represents statistical error data that was generated to help visualize both scenarios based on its distance error values. Scenario 2 has a maximum error value of approximately $15,29$ m, while scenario 3 has a maximum of approximately $15,23$ m. Minimum values were surprisingly low, $0,023$ m

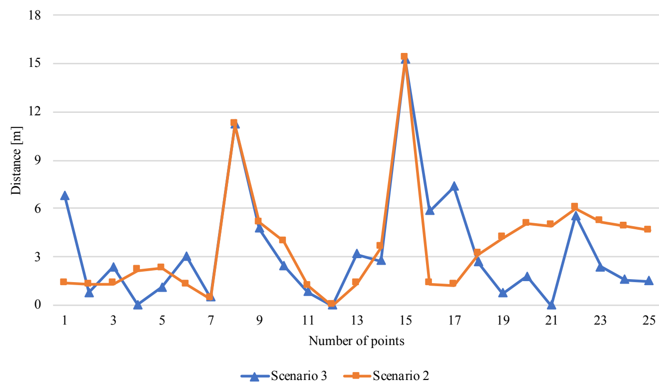

Fig. 6. Relative error on both scenarios.

TABLE IV
DISTANCE ERROR STATISTICAL DATA

Error	Maximum [m]	Minimum [m]	Mean [m]
Scenario 2	15,29	0,023	3,70
Scenario 3	15,23	0,017	3,38

and $0,017$ m, for scenarios 2 and 3, respectively. The mean error is approximately $3,7$ m for scenario 2 and $3,38$ m for scenario 3. This represents a small difference of approximately $0,32$ m.

According to the distance error statistics represented by Table IV, scenario 3 is the most accurate path when compared with theoretical path values, since it has the lowest values on minimum, maximum and mean error data.

V. CONCLUSIONS AND FUTURE WORK

This work test case scenarios and its results accomplished that it is impossible to use the frequency of the GNSS sensor (5 Hz) with the LoRaWAN™. The main reason was due to LoRaWAN™ gateway limitations and restrictions, such as it only allows one transmission per minute per device and a maximum payload length of 248 bytes. This could be caused by the number of devices connected to the gatway during test cases, but there is no information about this. In order to make transmission possible according to established conditions, it was necessary to compile all the information in a single string to be sent every minute. However, a higher sampling frequency means a greater string size, and at a sampling frequency of 5 Hz, the string size exceeded the maximum payload length allowed by LoRaWAN™ gateway. The maximum sampling frequency which string size didn't exceed maximum payload length was at $0,5$ Hz, which string size was 248 bytes. These limitation values were obtained after various trial-and-error earlier experimental tests. This also makes the PPK approach not to be the best solution for position corrections, because PPK systems need GNSS position readings at maximum sampling rate allowed by GNSS receiver sensor. Actually, these readings are possible by using SmartEverything Lion onboard GNSS sensor, which was the one used on test case. The only problem on achieving maximum sampling rate was the gateway limitations and restrictions, as stated before, making impossible to gather GNSS data at maximum sampling frequency, invalidating this solution on a PPK environment system.

Although, this paper test case scenarios showed the importance of testing the LoRaWAN™ gateway limitations and restrictions. Also, LoRaWAN™ communication protocol can be a suitable solution for future use case applications, considering other areas. With this research work, it was also a good challenge to reach maximum accuracy possible by optimizing the transmitted GNSS data type models. This research also demonstrated the importance of the efficiency of data transmission in IoT scenarios when using LoRaWAN™ communication protocol.

For future work, other communication protocols should be studied in the same scenarios in order to assess the correction of position during the course of the UAV.

REFERENCES

[1] A. E. Shouny, N. Yakoub, and M. Hosny, "Evaluating the Performance of Using PPK-GPS Technique in Producing Topographic Contour Map," *Marine Geodesy*, vol. 40, no. 4, pp. 224–238, 2017.

[2] D. Ershov, "Proving the Efficiency of PPK Drones for Agricultural Survey." [Online]. Available: https://emlid.com/proving-efficiency-ppk-drones-agricultural-survey/

[3] A. Gloria, F. Cercas, and N. Souto, "Comparison of communication protocols for low cost Internet of Things devices," in *2017 South Eastern European Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*. IEEE, 9 2017, pp. 1–6.

[4] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the Limits of LoRaWAN," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34–40, 2017.

[5] B. Crow, I. Widjaja, L. Kim, and P. Sakai, "IEEE 802.11 Wireless Local Area Networks," *IEEE Communications Magazine*, vol. 35, no. 9, pp. 116–126, 1997. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=620533

[6] E. Ferro and F. Potortì, "Bluetooth and Wi-Fi wireless protocols: A survey and a comparison," *IEEE Wireless Communications*, vol. 12, no. 1, pp. 12–26, 2005.

[7] S. Mathur, A. Sankar, P. Prasan, and B. Iannucci, "Energy Analysis of LoRaWAN Technology for Traffic Sensing Applications," *Intelligent Transportation Systems (ITS) World Congress*, 2017.

[8] N. Sornin, M. Luis, T. Eirich, T. Kramp, and O. Hersent, "LoRaWAN Specification," LoRa Alliance, Tech. Rep., 2015.

[9] RS Components, "Arrow SmartEverything LION IoT SoM Board." [Online]. Available: https://uk.rs-online.com/web/p/processor-microcontroller-development-kits/1248830/

[10] Laird, "LoRaWAN Keys and IDs Overview," pp. 1–6, 2015.

[11] "GPS Visualizer." [Online]. Available: http://www.gpsvisualizer.com/

[12] "BlueMM." [Online]. Available: http://bluemm.blogspot.com/2007/01/excel-formula-to-calculate-distance.html