Capture of UAVs through GPS spoofing Using Low-Cost SDR Platforms

João Gaspar[1] · Renato Ferreira[1] · Pedro Sebastião[1] · Nuno Souto[1]

Abstract
The increased use of Unmanned Aerial Vehicles (UAVs), better known as drones, by civilians has grown exponentially and their autonomous flight control systems have improved significantly, which has resulted in a greater number of accidents and dangerous situations. To help cope with this problem, in this paper, we address the use of low-cost programmable Software Defined Radio (SDR) platforms for simulating a Global Navigation Satellite System (GNSS), more specifically the Global Positioning System (GPS), in order to transmit false signals and induce a location error on the targeted GPS receiver. Using this approach, a defensive system was implemented which can divert, or even take control of unauthorized UAVs whose flight path depends on the information obtained by the GPS system.

Keywords Unmanned Aerial Vehicles, Global Navigation Satellite System, GPS Spoofing, Software Defined Radio

1 Introduction

Due to the difficult problem of dealing with unauthorized operations of unmanned aerial vehicles (UAVs) and the growing occurrence of incidents, especially involving flights in areas close to airports, military areas, restricted areas, or dangerous areas, some solutions have started to emerge. These include jammers [1], firearms and hawks trained for the purpose of "hunting" the UAVs [2]. Some products already appeared on the market like the KNOX [3] developed by MyDefence, but only comprise a jamming system solution approach. All these methods have limitations since they can lead to damage the device itself, injure the animal responsible for the "hunting", as well as putting at risk the personal safety of the citizens present on the site where the intercepted device may land uncontrollably. Another possible solution for UAV control in zones or situations described previously is the spoofing of the UAV command, regardless of whether it is controlled by satellite navigation, WiFi or other radio waves. However communication protocols can vary between different brand and models of UAVs which makes it more complicated to implement spoofing for all of them. It is, therefore, simpler to perform the spoofing of satellite navigation signals since the GPS signal are broadly used by most UAVs, especially for autonomous operations . There have been some studies and investigation in this area of spoofing GPS against UAVs. In [4], the authors studied and explored the vulnerabilities of GPS systems in drones in order to divert or gain control over the aircraft. Also, another example of exploiting the GPS vulnerabilities is the UnicornTeam, a team of security researchers whose main focus is the security of systems using radio technologies. This is a team that has been part of the DEF group with 23 vendors and proven with various approaches that it is possible to spoof a GPS receiver in [5].
Due to the its flexibility for multiple applications, there has been a growing interest in the use of Software Defined Radio (SDR) for implementing and testing radio systems. A SDR platform, such as the bladeRF used in the tests presented in this paper, is a radio communication system where components that where traditionally implemented in hardware (e.g. mixers, filters,

modulators/demodulators) can be developed in software using the right frameworks for each different brand and model of SDR.
In this paper, we describe the development of a mobile spoofing system which integrates low-cost SDR platforms and a software GPS signal simulator combined with a set of sensors to determine the unauthorized UAV location. The implemented system is capable of transmitting false GPS signals to redirect or even gain control of the vehicle flying over protected areas. For evaluating the behavior of the system's operation, several types of GPS receivers were tested as targets for the spoofed signal in different scenarios.
The remainder of the paper is organized as follows: Section 2 introduces the global satellite navigation systems with focus on the GPS system. Section 3 describes what is spoofing and presents some techniques used in the spoofing of GPS signals. Regarding the developed spoofing system, the description of its operation and its architecture are presented in Section 4. In section 5 describes the experimental tests using the system developed for spoofing different GPS receivers. Finally, in section 6 the conclusions are drawn.

2 Global Navigation Satellite System

GNSS systems have a high level of complexity because they comprise various subsystems working together. While the satellites are the more "visible" part of the system, terrestrial infrastructures are crucial for correct operation supporting necessary maintenance tasks of the satellites orbits. Users only have access to a radio link in the system, the downlink transmission from the satellites of the constellation. Since the downlink signal is transmitted in broadcast, there is no limit on the number of users of the system [6].

2.1 Global Positioning System

The GPS system is the only one explored in this paper since it was the first system to come into operation and, currently it is the most commonly used system.
Its architecture is divided into several segments: user segment, which consists of all types of GNSS GPS signal receivers, a space segment, which brings together all satellites constituting the constellation, and a ground segment, which is responsible for monitoring, controlling and updating stations [7], [8].

Ground segment
Main functions are:
• Monitor the satellites;
• Define the orbits for each satellite to predict the ephemeris and almanac data;
• Determine the altitude and location of each satellite and send the corrections to the satellites so they remain in the correct orbit [7], [8].

Space segment
Constellation base containing 24 satellites, consisting of six almost circular orbits with a slope of 55 ° referenced with the equatorial plane at an altitude of 20183 km. Each satellite can make a circle around the planet in exactly 11 hours 57 minutes and 58 seconds. This makes it possible to have four satellites in line of view in any position on the planet thus, enabling localization, under normal atmospheric conditions. The constellation was officially declared operational in 1995.
Main functions are:
• Receive from the ground segment the corrections of the orbits apply them;
• Transmit the GNSS signals [7], [8].

User segment
It consists of a wide variety of receivers, including military, mass-produced receivers for civil use and even for scientific purposes. Its main functions are:

• Receive signals corresponding to GNSS systems and evaluate their status;
• Perform measurements of propagation time;
• Perform measurements due to the Doppler effect;
• Calculate the location of the receiver;
• Calculate the speed of the terminal and provide time measurements [7], [8].

2.2 GPS Frequencies, Codes and Modulations

The GPS system has 3 frequency ranges, L1, L2, and L5 being L2 and mainly L5 frequencies still with some development [9]. In this paper, only the L1 frequency range was addressed.
L1 is the most commonly used worldwide GPS frequency range. It operates at a 1.57542GHz frequency and its access is by CDMA (more details are provided in Table 1-1). It contains three different signals: Coarse/Acquisition (C/A) code, P code, and M code.

C/A code: became the most adopted and important code, intended for civil use, and many solutions developed in the market to use the GPS system rely on this signal. It has a millisecond length at a chipping rate of 1,023 Mbps [10].
P code: is a precision code intended only for military applications. The P (Y) code is often employed in place of the P code when using anti-spoofing systems. The code features a seven day long length, with a chipping rate of 10.23Mbps and guarantees confidentiality and authentication [10].

Code M: is designed exclusively for military use and may eventually replace the P and P (Y) code. It has better features to resist jamming and guarantees better performance and more flexibility than the P (Y) code [10].

Table 1: GPS L1 technical features [9]

Service	C/A	P(Y) code	M code
Modulation	BPSK(1)	BPSK(10)	$BOC_{sin}(10,5)$
Code Length	1023	6.19×10^{12}	_____

It can be concluded that GPS, in the L1 frequency range, is divided into two main types of transmitted GPS signals:

• Open signals for civil use;

• More robust and more accurate signals, for military use.

For this purpose, the codes described above remain in use and the localization is divided into two services: Standard Positioning Service (SPS) and Precise Positioning Service (PPS), which correspond respectively to the two different types of signals.
SPS is a service that can be accessed by any normal (civil) user. It is based on the C/A code sequence. The PPS service is only accessible to authorized users (military) and it not only uses C/A code but also P-code. This allows a greater accuracy of the location on the globe.

3 Spoofing

Spoofing, in general, is a fraudulent or malicious practice in which communication is sent from an unknown source, disguised as a source known to the receiver. The use of spoofing is more common in mechanisms and communication networks that do not have a high level of security. This is the case of the civil Global Positioning System (GPS) signal, which does not have any type of encryption or authentication to protect or to prove that the signal comes from a reliable source or the non-occurrence of repudiation of the signal. So, to accomplish spoofing and deceive a GPS receiver, it is necessary to simulate GPS signals as if they came from real satellites.
All devices that use radio frequency for communication have the vulnerability that the information transmitted is available to everyone within the range of the transmission.

3.1 Various spoofing techniques

Most of the systems using GPS signal receivers are vulnerable to the spoofing techniques described below:

Simple spoofing: technique capable of generating false Global Navigation Satellite System (GNSS) signals. It can be put into practice using:

- Low-cost hardware to receive and reproduce GNSS signals. Custom signal simulators can be inserted into the configuration to control and modify some of the transmission parameters [11];
- Commercial hardware that is usually expensive and more complex to manipulate, but often with greater capacities for the processing and transmission of electromagnetic signals [11].

Intermediate spoofing: in this case, the attacker synchronously generates false signals while simultaneously attempting to attack each channel of the target receiver by performing code phase alignment between false and genuine incoming signals [12].

Spoofing with multiple transmitting antennas: advanced technique, used mainly against multiple antenna receivers, in which each transmitting antenna of the attacker combines with a corresponding receiving antenna in the victim [13].

Spoofing with high gain antennas: enhanced attack based on the use of antennas with enough gain to separate GNSS signals from noise, including, for example, unknown or encrypted code chips [14].

Sophisticated spoofing: can be performed by a set of coordinated and synchronized attackers, capable of attacking the victim's receiver in an organized manner. In addition, they have three-dimensional position information about the phase centers of their antennas and the phase center of the victim's antenna, thus overcoming complex countermeasures, such as those based on the estimation of the angle of arrival [12].

4 Mobile spoofing system

The spoofing system developed and described in this paper operates using an open hardware electronic prototyping platform, sensors, an SDR module and a System on a Chip (SoC) as the central processor of the system. This system adopts a simple spoofing technique, which can generate and transmit false GPS signals. However, it takes into account the current location of the UAV and employs a directive antenna for focusing the transmission on the intended target, making the system more sophisticated and more difficult to detect by the UAV control station.

4.1 UAV Location Determination

To determinate the location of the UAV, the implemented system integrates three sensors and a receiver, namely:

- Lidar sensor Lidar Lite v3 [15] (measurement of UAV distance);
- 3D accelerometer MPU6050 [16] (tilt/pitch measurements);
- e-compass LSM303D [17] (measuring system orientation);
- GPS signal receiver [18] (detection the system positioning).

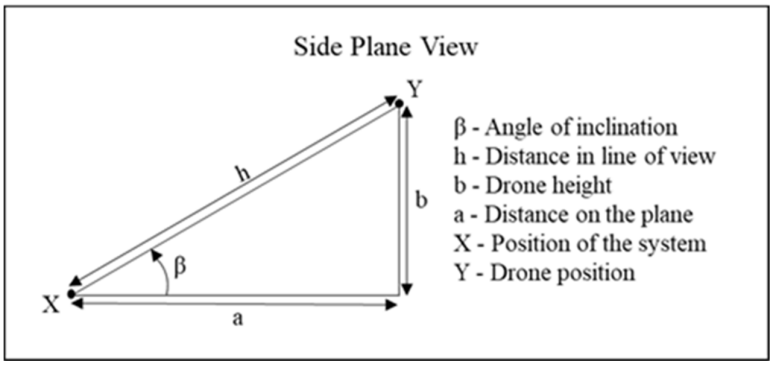

Figure 1: Side plane view analytical calculations

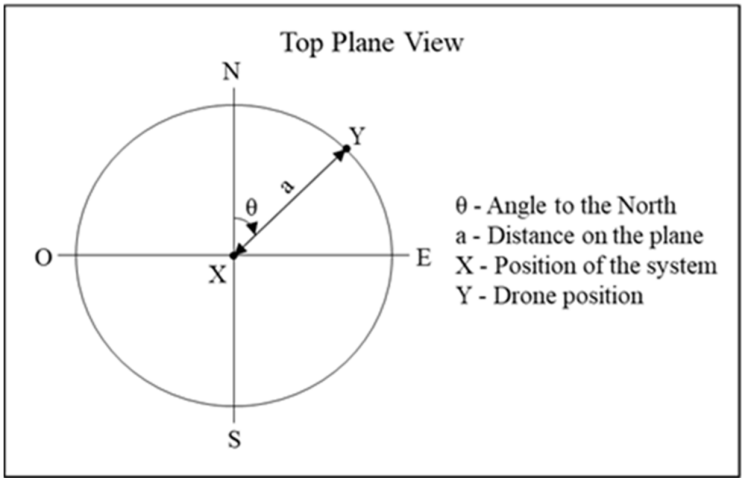

Figure 2: Top plane view analytical calculations

By associating the measurements of the sensors with the angles defined in Figure 1 and Figure 2 the location is determined as follows:

- UAV distance in line of sight: Sensor Lidar measurement = h;
- The tilt angle with the horizontal plane: 3D Accelerometer measurement = β;
- System orientation angle: Magnetometer measurement = θ;
- System location: GPS receiver = X.

After obtaining the values of the sensors and receiver, 'a' and 'b' are calculated using simple trigonometric equations as

$$a = \cos(\beta) \times h \tag{1}$$

$$b = \cos(\beta) \times h. \tag{2}$$

With the result of the UAV distance in the horizontal plane (variable 'a') associated with the value exported from the magnetometer (variable 'θ'), the mean value of the radius of planet Earth [19] and knowing the location of the system itself (variable 'X'), the location of the UAV's (variable 'Y') latitude and longitude can be determined using [20].

$$\text{LatY} = \sin^{-1}(\sin(\text{LatX}) \times \cos\left(\frac{\frac{a}{1000}}{\text{R}_{\text{earth}}}\right) + \cos(\text{LatX}) \times \sin\left(\frac{\frac{a}{1000}}{\text{R}_{\text{earth}}}\right) \times \cos(\theta)) \quad (3)$$

$$LongY = LongX + tan^{-1}\left(\frac{cos\left(\frac{\frac{a}{1000}}{R_{earth}}\right) - sin(LatX) \times sin(LatY)}{sin(\theta) \times sin\left(\frac{\frac{a}{1000}}{R_{earth}}\right) \times cos(LatX)}\right) \quad (4)$$

where the following definitions are adopted

- LatX - Latitude value of system location;
- LongX - Longitude value of system location;
- LatY - Latitude value of UAV location: LatY;
- LongY - Longitude value of UAV location: LongY;
- Rearth - Planet Earth radius (≈6378.137 km).

Note that the value of '*a*' and 'Rearth' need to be on the same scale (kilometers) and all angles must be in radians, not degrees.

4.2 Deviation of UAV

After acquiring the location of the UAV through the process previously described, the next step of the GPS spoofing system is reached, namely the generation of the fake signal. For the transmission of false GPS signals we used the bladeRF x40 SDR platform. It presents a basic radio architecture, but it is capable of encompassing modulation techniques and basic telecommunications coding schemes. It has USB 3.0 communication capability and a fully programmable FPGA chip for faster system development [21]. The choice of the bladeRF was made taking into account its low energy consumption and its versatility [22]. To implement the spoofing step, we adopted the free available online software, bladeGPS simulator, which was developed by OSQZSS in Japan and is capable of constructing and simulating real GPS signals, available link as a note below. Looking at the different functionalities available in the bladeGPS simulator, there is one that can be easily exploited for implementing a spoofing system.
In fact, one of the functions provided by the bladeGPS simulator is the ability to use NMEA messages, marked in red in Figure 3, for the dynamic simulation of GPS signals. This enables the generation of not only static localizations but also trajectories and allows a simple way to construct GPS messages equal to the real ones, thus, spoofing the UAV current location.

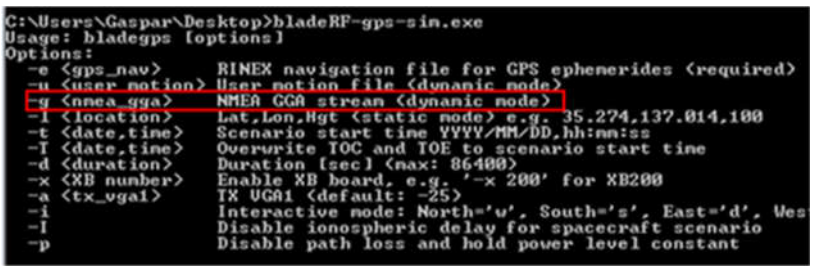

Figure 3: BladeGPS software interface

The idea of messages in NMEA format is to send a data line called a sentence that is totally independent of the previous and posterior lines. The information in the sentence is formatted according to the category of device that will receive them, indicated by a two-letter prefix. In the case of GPS receivers, the prefix is Global Positioning (GP) [23], [24] and for GPS fix location it is completed with another three-letters prefix, GGA, meaning Global Positioning System Fix Data.
NMEA messages were developed by the National Marine Electronics Association, which develops specifications that define the interface between various marine systems and electronic

Note: https://github.com/osqzss/gps-sdr-sim

equipment. The communication for GPS signal receivers is defined in these specifications [15]. Most computer programs that provide real-time positioning information understand and expect to receive information in NMEA format.
Each sentence starts with the character '$', ending with '*' and the value of the checksum (represented by two hexadecimal numbers). The checksum is calculated with an XOR operation of all characters between '$' and '*'. All information is contained in a single line with the various data separated by commas and represented in ASCII text. It can never exceed 80 characters per sentence. The first data consists of a code name that defines the type of data found in the sentence. Each data type has its own interpretation and is defined in the NMEA standard. The GGA sentence provides GPS correction data [25]. An example of a GPGGA sentence with the definition of each data field is shown in Figure 4.

$GPGGA,090000.00,3845.19975115,N,00910.36577529,W,1,05,2.87,160.00,M,-21.3213,M,,*6F

Where:

GGA	Global Positioning System Fix Data
090000.00	Fix taken at 09:00:00 UTC
3845.19975115,N	Latitude 38° 45.1997' N
00910.36577529,W	Longitude 9° 10.3657' W
1	Fix quality: 0 = invalid
	1 = GPS fix (SPS)
	2 = DGPS fix
	3 = PPS fix
	4 = Real Time Kinematic
	5 = Float RTK
	6 = estimated (dead reckoning)
	7 = Manual input mode
	8 = Simulation mode
05	Number of satellites being tracked
2.87	Horizontal dilution of position
160	Altitude, Meters, above mean sea level
00	Height of geoid (mean sea level) above WGS84 ellipsoid
(empty field)	time in seconds since last DGPS update
(empty field)	DGPS station ID number
*6F	the checksum data, always begins with *

Figure 4: NMEA example message

In this way, as we already know how the NMEA messages are created and the current location of the UAV, the process of simulating the GPS signals for the deviation of the UAV position can be synthesized in the elaboration of an NMEA message with a sentence sequence that indicates a false current location to the vehicle. This can cause the vehicle to try to correct its present position and thus change its actual position to a position outside the restricted area, as in Figure 5.

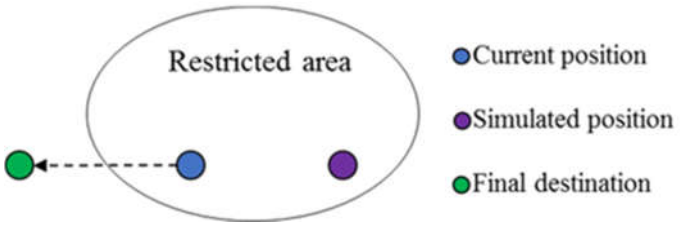

Figure 5: Example of illustration of static spoofing

The sentence formation was also constructed dynamically, i.e., simulating a moving location, causing the UAV to correct its course in continuous displacement, but with a route that will direct it to the landing area. To do this, it is necessary to determinate two consecutive locations of the UAV through the measurements of the sensors and receiver of the system, as previously explained in Figure 1 and Figure 2. With these two locations (red markings in Figure 7) and the time interval between them, the system can determine not only the direction of the UAV path as well as its speed.

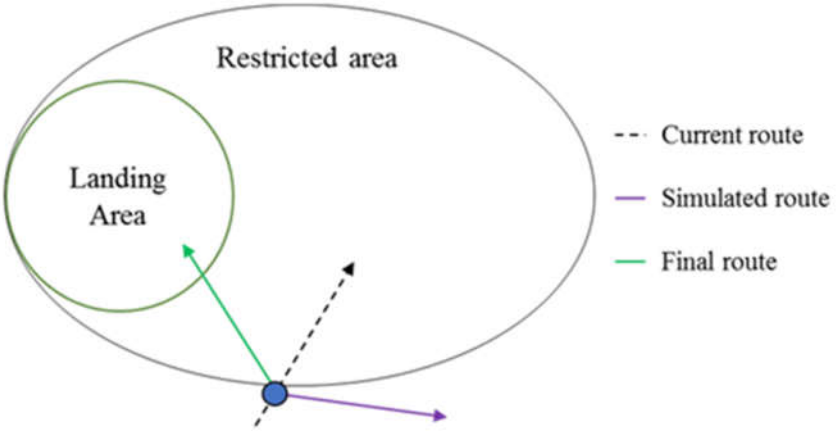

Figure 6: Example of illustration of dynamic spoofing

It was defined the Maps JavaScript API as the updated map source which allows to use existing functions and develop a graphical interface that presents the results of the system and its current state. Using the API, it can be calculated and traced the "Simulated Route" necessary to drift the UAV to the landing area, with the following steps:

- Distance Computation: calculate the distance between the second location of the UAV, obtained from the sensors in the system, and the landing area. It is exemplified as the "Final route" in Figure 7;
- Heading Determination: determine the angles α_1 and α_2 shown in Figure 7, namely, the angle of the UAV's original course line ("Current route" in Figure 7) and the angle of the line between the UAV and the landing area ("Final route" in Figure 7). These angles are measured using the north bearing as a reference;
- Offset Determination: Given angle α_3 (in Figure 7), the location of origin (last location of the UAV through the sensors of the system), and the distance to the Landing Area it is possible to compute the "Simulated route" as in Figure 7. Note that angle α_3 is easily obtained as

$$\alpha_3 = (\alpha_1 - \alpha_2) + \alpha_1 \quad (5)$$

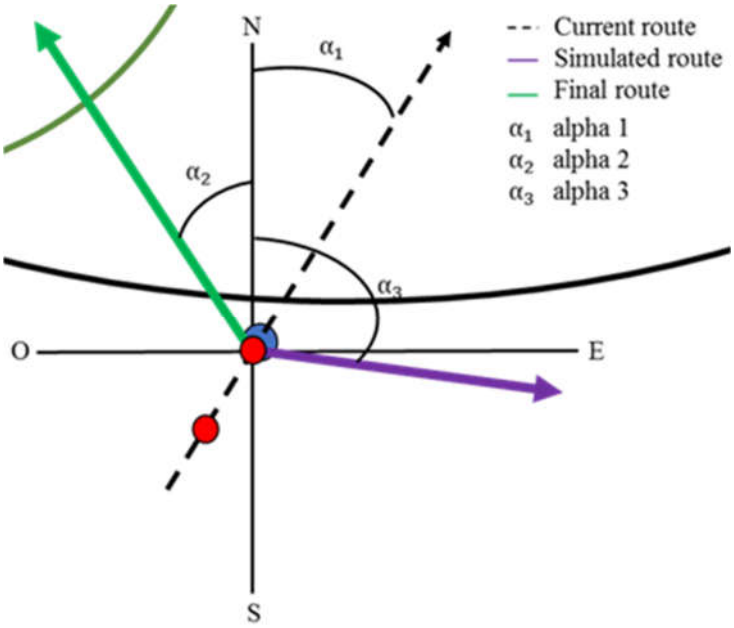

Figure 7: Angles and computations for the dynamic spoofing

4.3 Architecture

A general scheme of the whole mobile spoofing system is represented in Figure 8, for an easier perception of its operation. It is divided into different blocks which we describe next. The corresponding implementation adopted for the experimental tests are shown in Figure 9.

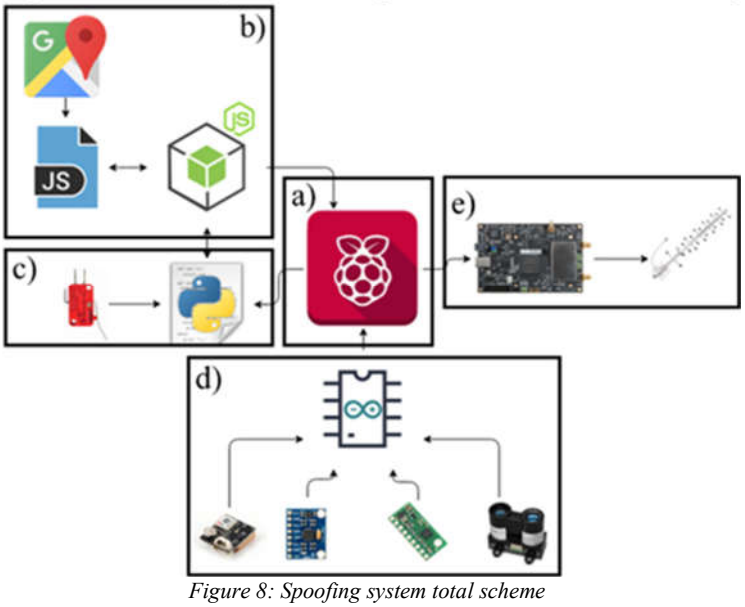

Figure 8: Spoofing system total scheme

- a): A Raspberry Pi used as the central controller of the system. It relied on a Linux Ubuntu distribution install with all scripts and software saved in memory. It distributes the commands and tasks to other subsystems and scripts;
- b): In this block there are a set of scripts responsible to download the maps and construct the NMEA messages (developed in JavaScript and implemented with NodeJS). It uses the framework Electron of NodeJS to implement the system as a desktop application that runs on boot;

- c): Comprises a script (developed and implemented in Python) responsible to redirect the sensors values to the NodeJS application and read the switches state to trigger the system to initiate or stop the transmission of the spoofing signals;
- d): Corresponds to the sensing unit. It was implemented using an Arduino Uno board with all the sensors and receiver mentioned and described previously in order to estimate the location of the UAV (developed and implemented in C++). The communication with the sensors is established through i2c bus, with all the sensor data transferred to block a) using Universal Serial Bus (USB) interface;
- e): Output block that is in charge of the transmission of the false GPS signals. It is supported by a bladeRF SDR board connected to a YAGI antenna with high transmission gain and directivity.

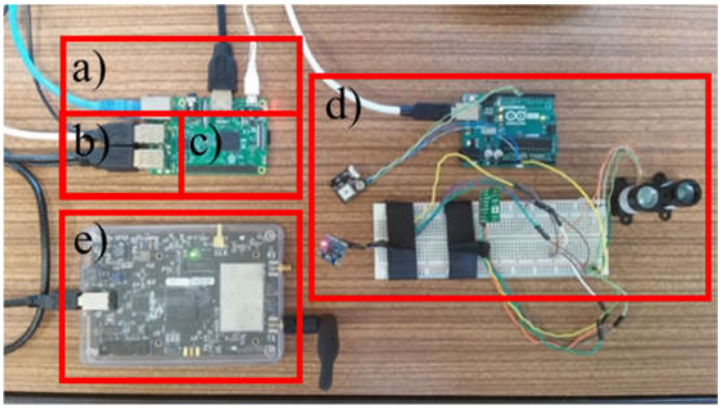

Figure 9: Spoofing system in a laboratory environment

5 Experimental Results

Several tests were performed on the sensors used for estimating the UAV location in order to gauge their limitations and possible influences on the final results. Spoofing tests were also made using the overall system in an indoor and outdoor environment.
Three different types of GPS receivers were used as targets:

- smartphone;
- u-blox M8 GNSS Evaluation Kit;
- u-blox MAX-7Q receiver.

5.1.1 Sensor tests

Measurement accuracy tests of the lidar, magnetometer, and accelerometer were accomplished in laboratory. These tests were designed to individually evaluate each of the sensors in question, determining their possible read errors and help the integration into the overall system.

LIDAR Lite v3

The laboratory tests performed on the sensor were precision measurement tests. For the test, a measuring tape of 10 meters was used and measurements were taken spaced by one meter. As the distance increases, an increase in the measurement error also occurs, as seen in the test results in Table 1 and Figure 10, which shows a linear upward trend line. Because the error values are not very high, they can not significantly influence the final measurements for determining the location of the drone do to the original tolerance error in the real GPS system.

Table 2: Error values measured by the Lidar sensor

Distance[m]	1	2	3	4	5	6	7	8	9	10
Error[m]	0.026	0.031	0.017	0.02	0.08	0.12	0.1	0.14	0.11	0.13

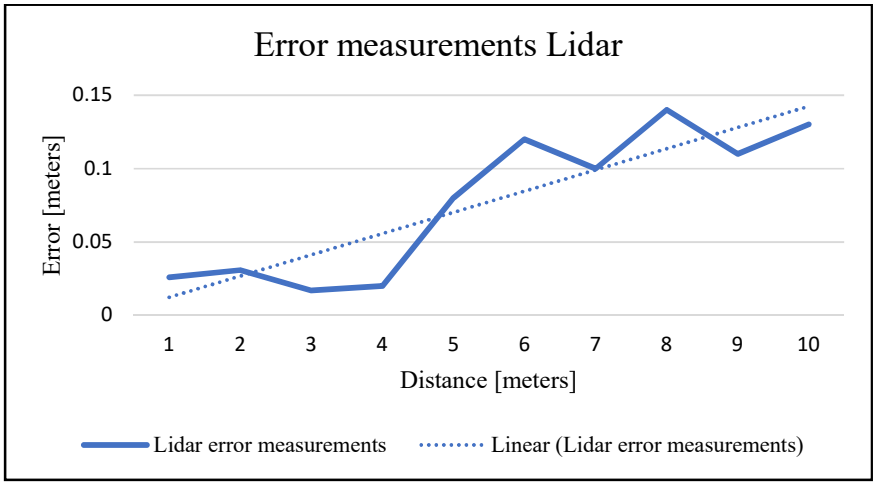

Figure 10: Precision measuring chart of the Lidar sensor

MPU6050

The precision test was developed using a smartphone and its gyroscope. Encapsulating the MPU6050 sensor to the smartphone ,measurements were taken with a five-degree interval, in a range of 0º to 90º.
Given the values indicated by the smartphone and the values measured by the MPU6050 sensor, we take 15 samples and calculated an average for every measurements angle. The difference between both values of the platforms was calculated and are presented in Table 2. The behavior of the error with the variation of the angle is also shown in the graph of Figure 11.

Table 3: Error values measured by the MPU6050 sensor

Angle[°]	0	5	10	15	20	25	30	35	40	45	50	55
Error[°]	0.02	0.37	0.49	0.1	0.12	0.28	0.41	0.25	0.52	0.32	0.48	0.42

Angle[°]	60	65	70	75	80	85	90
Error[°]	0.74	0.54	0.22	0.18	0.24	0.36	0.08

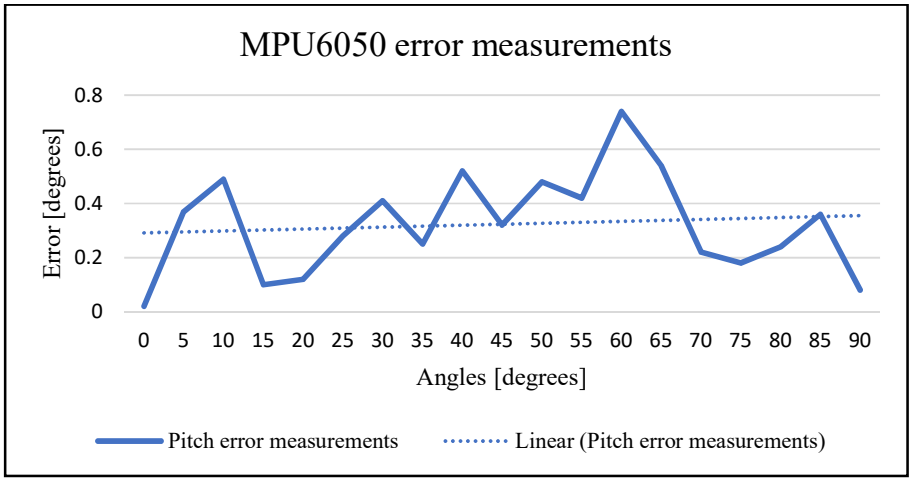

Figure 11: Precision measuring chart of the MPU6050 sensor

It has been found that with increasing angle, there is also a slight increase in measurement error. This can be seen from the values in Table 2 and the chart of Figure 11, which shows a linear upward trend line of very low slope. Therefore, these errors present a negligible influence on the final measurements to determine the location of the drone.

LSM303D

A precision test was developed using a smartphone and its magnetometer. Accommodating the lsm303d sensor to the smartphone, measurements were taken with a range of 10º, in a range of 0º to 360º.
Using the values provided by the smartphone and the values measured by the sensor lsm303d we take 15 samples and calculated an average for every measurements angle. The difference between the values of both platforms were calculated. The resulting error and its variation with the angle is presented in Table 4 and Figure 12.

Table 4: Error values measured by the LSM303D sensor

Angle[°]	0	10	20	30	40	50	60	70	80	90	100	110	120
Error[°]	1.03	1.89	1.12	2.01	2.32	1.65	2.78	2.65	1.98	1.21	1.54	1.3	1.78

Angle[°]	130	140	150	160	170	180	190	200	210	220	230	240	250
Error[°]	1.59	2.06	0.83	1.45	1.81	1.08	1.09	2.04	1.64	1.12	1.95	1.11	2.16

Angle[°]	260	270	280	290	300	310	320	330	340	350
Error[°]	1.27	1.38	1.34	2.42	3.48	1.84	2.09	1.54	1.62	1.65

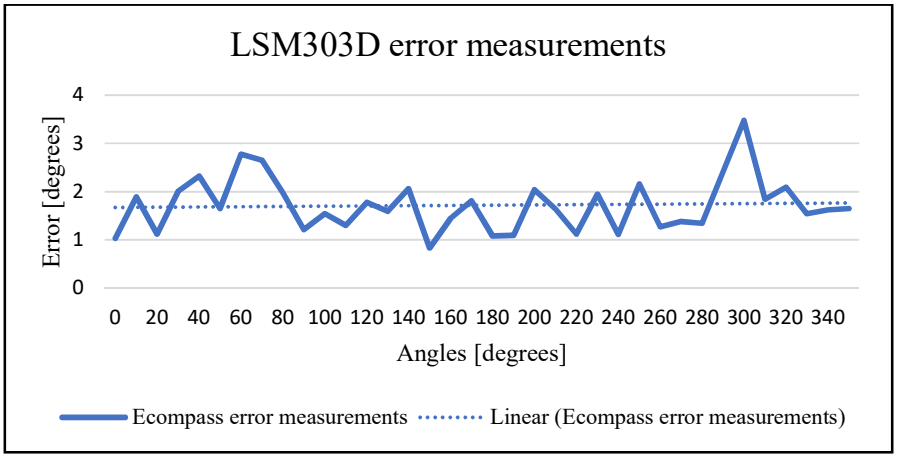

Figure 12: Precision measuring chart of the LSM303D sensor

By varying the angle, there is no increase in the measurement error, as can be seen in the values in Table 3 and in the graph of Figure 12. A horizontal linear trend line is shown, demonstrating that the measured errors vary, but with a certain coherence in the differences of values between platforms. The recorded error values can cause a larger influence than the previous sensors in the determination of the location of the drone.

5.1.2 Indoor tests

The first phase of tests were carried out inside a building, that is, without the influence of actual GPS signals. This allows testing the response of the various target receivers to the spoofing system when they have no previously acquired location.

Smartphone

The tests have been made with a smartphone to verify that many of the GPS receivers installed on these devices are vulnerable to spoofing attacks. The smartphone selected as a target for these tests was used as the LG L90 model. The GPSTest application [26], Figure 10, was installed in the device, which allows real-time visualization of which satellites are visible, GPS signal power levels and location in the world map.

Figure 10: Android GPSTest application

Using the system developed and described in Section 4 it was possible to simulate, with coordinates defined by the user, the location of the smartphone in relation to its real position. As shown in Figure 11 on the left, several signals from satellites with good SNR are detected. On the center some of the simulated satellites are shown in the line of sight and, to the right, the false location is shown through the latitude, longitude and with the red marker on the world map. Note that the application estimated the location on Caracas Venezuela, even though the true location was Lisbon Portugal. With this test, it was verified that it is possible to deceive the location of a GPS receiver installed in a smartphone in an indoor environment.

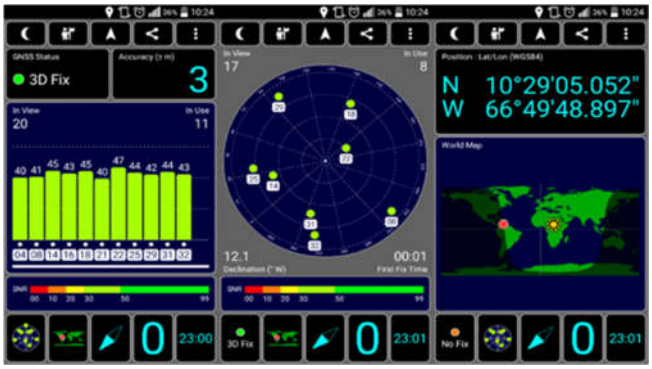

Figure 11: Smartphone GPS spoofing indoor

u-blox M8 GNSS Evaluation Kit

for the second target was the u-blox receiver M8 GNSS Evaluation Kit, Figure 12. The M8 u-blox evaluation kit allows a simple evaluation of positioning technologies. It features an integrated USB interface that provides power, eliminating the need for external power supply while supporting high-speed data transfer. The receiver was used with a computer via the USB interface in conjunction with the u-center software, which is a powerful tool for evaluating, performing and configuring u-blox GNSS receivers.

Figure 12: u-blox M8 GNSS Evaluation Kit

It was possible to simulate, with user-defined coordinates, the location of the u-blox receiver M8 GNSS Evaluation Kit in relation to its actual position. As shown in the lower part of Figure 13, several signals from GPS satellites with good signal noise ratio (SNR) are detected in the line of sight in the constellation. At the top of the monitor, the simulated location is shown on the world map through a green marker. In this case, the location obtained by the u-center was Washington DC United States of America, but the true one was Lisbon Portugal. With this experience, it has been found that it is possible to mislead the location of a GPS receiver u-blox prepared to evaluate location systems in an indoor environment.

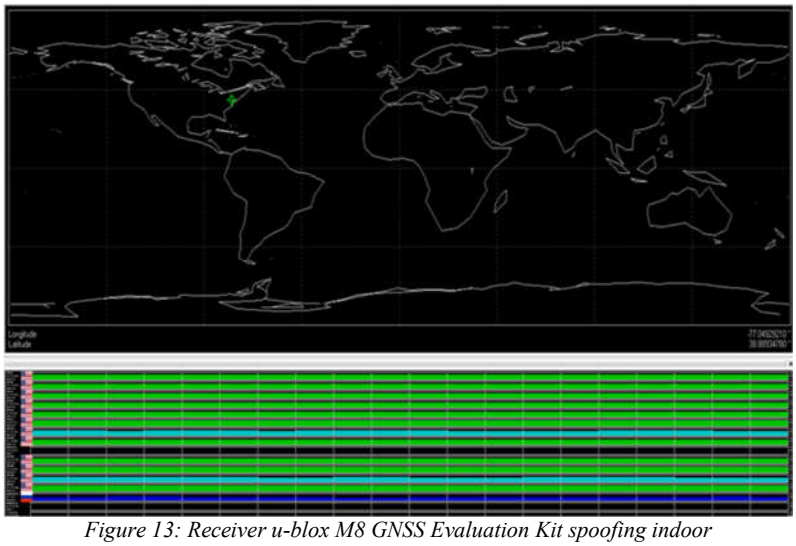

Figure 13: Receiver u-blox M8 GNSS Evaluation Kit spoofing indoor

u-blox MAX-7Q

Considering that the u-blox MAX-7Q receiver, Figure 14, is widely used in several types of terrestrial, aerial and aquatic drones for supporting autonomous missions, it was also tested as a target for the GPS spoofing. The receiver was used in conjunction with an Arduino, in order to communicate its location through the Arduino serial interface and enabling the visualization of the information.

Figure 14: GPS receiver u-blox MAX-7Q

In this experiment, this type of receiver did not present any type of resistance when receiving fake GPS signals. As can be seen in Figure 15, the receiver identifies its location with the latitude and longitude values of Pyongyang Korea of the North, which do not correspond to the true ones 38.74673, -9.15274 (Lisbon Portugal). With this test, it was verified that in an indoor environment it is possible to easily deceive the location of a u-blox GPS receiver used in many drones.

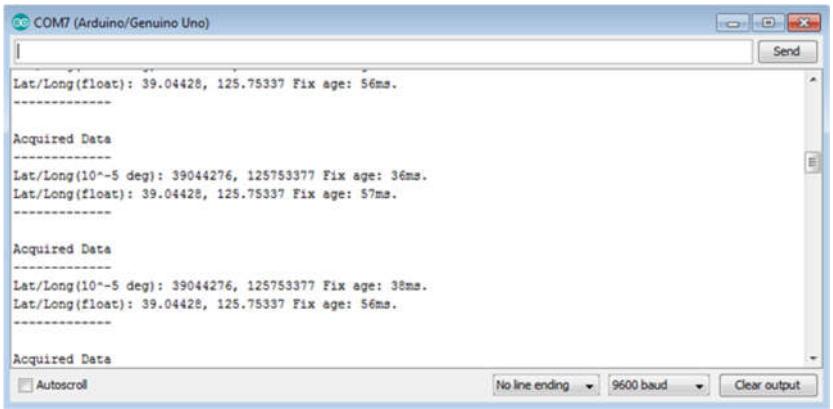

Figure 15: Arduino serial interface spoofing indoor

5.1.3 Outdoor tests

The following outdoor tests were carried out under the influence of real GPS signals. This allows the evaluation of the spoofing system performance in a scenario where the receiver already has a pre-acquired location through real GPS signals.

Smartphone

Outdoor smartphone tests were carried out to verify that many of the GPS receivers installed in these devices are vulnerable to spoofing attacks, even after they already have obtained a location through real GPS. Once again, the smartphone model used as the target was the LG L90.

For these tests, after the smartphone acquired true location, the spoofing system started to transmit the fake GPS signals. It was observed that after about 3 minutes the system lost location, taking into account that it detected another GPS signal (false GPS signal transmitted by the bladeRF platform), and then accepted the false GPS signal possibly because it had better SNR. Under these conditions, and with the results described above, it was possible to simulate, with user-defined coordinates, the location of the smartphone in relation to its actual position. As shown in Figure 16, at the left side several satellite signals with good SNR are detected. In the center are shown some of the simulated satellites and real satellites of the constellation. On the right, it is shown as a blue dot the wrongly induced location of the smartphone in the GoogleMaps application. The true position of the smartphone corresponds to the red dot. With this test, it was verified that it is possible to deceive the location of a GPS receiver installed in a smartphone in an outdoor environment, even under the influence of real GPS signals.

Figure 16: Smartphone GPS spoofing outdoor

u-blox M8 GNSS Evaluation Kit

Following a similar test approach, after the u-blox receiver, M8 GNSS Evaluation Kit got its location, the transmission of false GPS signals was started. After about 2 minutes the receiver had already detected the new GPS signals, but only after about 30 minutes, it accepts the GPS signals created by the bladeRF platform.

Treating itself as a receiver used to evaluate and analyze the performance of GNSS systems, it has features that make it less susceptible to spoofing and jamming attacks. Hence its behavior makes it more difficult to accept the false GPS signals transmitted.

As shown in Figure 17, at the bottom of the monitor are shown some of the simulated and real satellites in line of sight. On the right side of the monitor is presented the world map with the simulated location represented as a green marker, while the actual location of the receiver was the red dot.

With this experiment, it was observed that it is possible to mislead the location of a GPS receiver u-blox M8 in an outdoor environment already with the defined location, but for the purpose of the developed system, the time it takes to accept the signals would be a critical point for the spoofing system. Possibly using jamming techniques before beginning the spoofing transmission would be a good option to speed up the process of deceiving the receiver with false GPS signals.

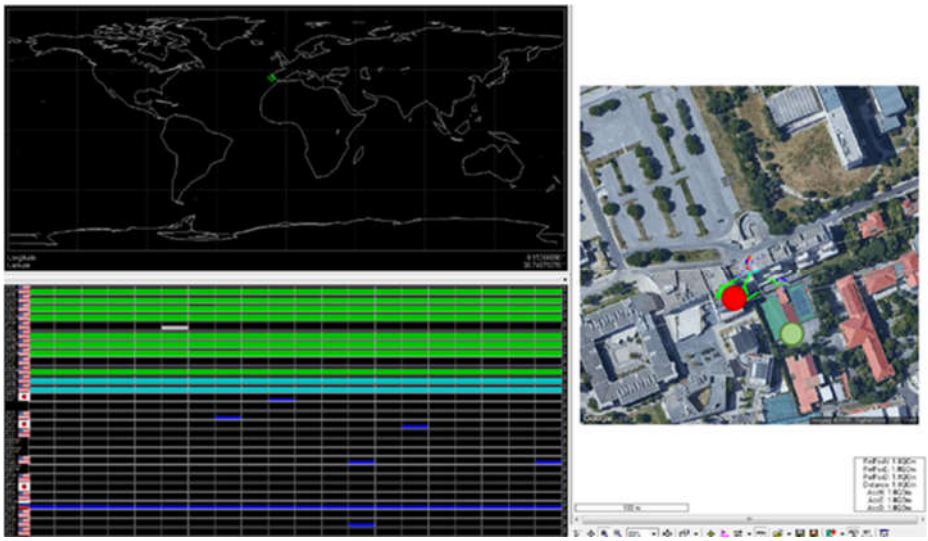

Figure 17: Receiver u-blox M8 GNSS Evaluation Kit spoofing outdoor

u-blox MAX-7Q

Finally, the u-blox receiver MAX-7Q was also tested in an outdoor environment where it already had acquired location through real GPS signals. It was observed that after 2 to 3 minutes it did not show any resistance when receiving the false GPS signals and accepted the false induced location. As can be seen in Figure 18, the receiver changes its location (marked in red latitude and longitude) according to the values entered in the bladeGPS software, Figure 19, that is, its true location is not the one displayed by the second values of latitude and longitude in the Arduino serial, but by the first one, Figure 18.

With this test, it was found that even in an outdoor environment it is possible to induce a wrong location on a u-blox GPS receiver which is widely used in drones.

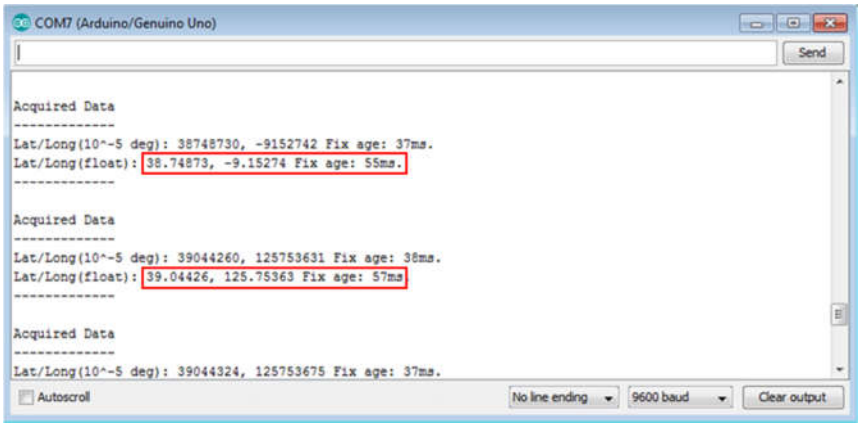

Figure 18: Arduino serial interface spoofing outdoor

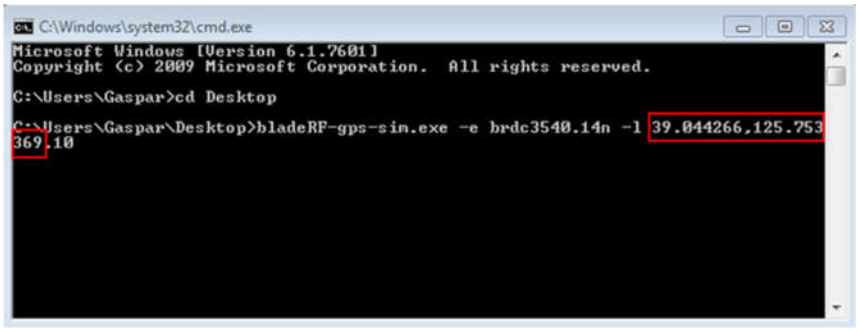

Figure 19: bladeGPS software location simulated

6 Conclusions

Using spoofing techniques, it is possible to recreate signals identical to the actual signals of existing systems, which makes it possible to elaborate advanced techniques of attacks that can be even capable of blocking the communications of a system.
In this paper, we described a possible elaboration of a portable system capable of diverting unauthorized UAVs using GPS spoofing techniques. The implemented system is based on flexible low-cost SDR equipment which is capable to transmit, receive, record and reproduce any radio communication systems. The development of the GPS spoofing system has proven that with a set of sensors, some analytic calculations and low-cost SDR equipment the GPS receivers do not have mechanisms protecting against spoofing and that it is possible to use a vulnerability of the GPS system to create something with practical applicability.

7 Acknowledgments

This work was funded by FCT/MEC through national funds and co-funded by FEDER – PT2020 partnership agreement under the project UID/EEA/50008/2019.

References

[1] M. Kratky and V. Minarik, "The non-destructive methods of fight against UAVs," *ICMT 2017 - 6th Int. Conf. Mil. Technol.*, pp. 690–694, 2017.
[2] The Guardian, "Eagle-eyed: Dutch police to train birds to take down unauthorised drones | World news | The Guardian," 2016. [Online]. Available: https://www.theguardian.com/world/2016/feb/01/dutch-netherlands-police-birds-unauthorized-drones. [Accessed: 08-Nov-2018].
[3] MyDefence, "KNOX Anti-Drone Solution - MyDefence Communication," 2017. [Online]. Available: https://mydefence.dk/anti-drone-solutions/knox-anti-drone-solution/. [Accessed: 26-Feb-2019].
[4] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via GPS spoofing," *J. F. Robot.*, vol. 31, no. 4, pp. 617–636, 2014.
[5] H. Lin and Y. Qing, "GPS SPOOFING Low-cost GPS simulator," 2015.
[6] A. Oxley, *Introduction to GPS*. 2017.
[7] N. Samama, "GNSS System Descriptions," in *Global Positioning*, John Wiley & Sons, Inc., 2007, pp. 131–161.
[8] GPS.gov, "NAVSTAR GPS Space Segment/Navigation User Segment L1C Interfaces IS-GPS-800," pp. 1–121, 2018.
[9] N. Samama, "Development, Deployment, and Current Status of Satellite-Based Navigation Systems," in *Global Positioning*, John Wiley & Sons, Inc., 2007, pp. 57–93.
[10] E. D. Kaplan, *Understanding GPS: Principles and Applications-2nd Edition*, 2nd ed. 2006.
[11] F. Dovis, *GNSS Interference Threats & Countermeasures*. Artech House, 2015.
[12] B. M. Ledvina, W. J. Bencze, B. Galusha, and I. Miller, "An In-Line Anti-Spoofing Device for Legacy Civil GPS Receivers," *Proceedings of the 2010 International Technical Meeting of The Institute of Navigation*, San Diego, CA, pp. 689–712, Jan-2010.
[13] M. L. Psiaki and T. E. Humphreys, "GNSS Spoofing and Detection," *Proc. IEEE*, vol. 104, no. 6, pp. 1258–1270, 2016.
[14] M. G. Kuhn, "An asymmetric security mechanism for navigation signals," *Int. Work. Inf. Hiding, IH*, vol. 3200, pp. 239–252, 2005.
[15] L. Lite and T. Specifications, "Table of Contents Lidar Lite v3 Operation Manual and Technical Specifications Wiring Harness," pp. 1–14, 2016.
[16] B. Ave, D. Number, and R. Date, "MPU-6000 and MPU-6050 Product Specification," vol. 1, no. 408, 2012.
[17] E. F. Ecopack, T. Lsm, T. Lsm, and T. Lsm, "3D accelerometer and 3D magnetometer

module," no. June, pp. 1–54, 2012.
[18] u-blox, "MAX-7 series | u-blox." [Online]. Available: https://www.u-blox.com/en/product/max-7-series. [Accessed: 30-Jul-2018].
[19] Dr. David and R. Williams, "Earth Fact Sheet," *NASA*, 2017. [Online]. Available: https://nssdc.gsfc.nasa.gov/planetary/factsheet/earthfact.html. [Accessed: 01-Aug-2018].
[20] C. Veness, "Calculate distance and bearing between two Latitude/Longitude points using haversine formula in JavaScript," *MIT License*, 2017. [Online]. Available: https://www.movable-type.co.uk/scripts/latlong.html. [Accessed: 21-Sep-2018].
[21] bladeRF Naund, "bladeRF x40 | Nuand -." [Online]. Available: https://www.nuand.com/blog/product/bladerf-x40/. [Accessed: 31-Jul-2018].
[22] Nuand, "bladeRF Power Consumption," 2017. [Online]. Available: https://github.com/Nuand/bladeRF/wiki/bladeRF-Power-Consumption. [Accessed: 18-Sep-2018].
[23] Dale DePriest, "NMEA data." [Online]. Available: https://www.gpsinformation.org/dale/nmea.htm. [Accessed: 17-Sep-2018].
[24] B. Park, J. Lee, Y. Kim, H. Yun, and C. Kee, "DGPS enhancement to GPS NMEA output data: DGPS by correction projection to position-domain," *J. Navig.*, 2013.
[25] G. P. S. C. Page, "GPS - NMEA sentence information," 2011. [Online]. Available: http://home.mira.net/~gnb/gps/nmea.html. [Accessed: 18-May-2018].
[26] "GPS Test – Apps no Google Play." [Online]. Available: https://play.google.com/store/apps/details?id=com.chartcross.gpstest&hl=pt_BR. [Accessed: 30-Aug-2018].

João Gaspar is a PhD student in Information Science and Technology at ISCTE-IUL. He is developing his thesis in the field of telecommunications studying and evaluating new strategies to protect areas of unauthorized drones through communications spoofing and beamforming.

Renato Ferreira graduated in electrical engineering and telecommunications in 2016, at the Polytechnic Institute of Castelo Branco, Portugal, and obtained his. Master MSc degree in telecommunications and computer engineering in 2018, at ISCTE - Instituto Universitário de Lisboa, Portugal. In Since 2018 he is a doctoral PhD student at ISCTE-IUL in Information Science and Technology at ISCTE-IUL. He is developing his doctoral thesis in the field of telecommunications studying and evaluating efficient techniques for neutralization of communications and radionavigation of multiple unmanned vehicles.

Pedro Sebastião is currently an assistant professor at the ISCTE-IUL faculty belonging to the Department of Information Sciences and Technologies. He is a researcher integrated in the Telecommunications Institute and is part of the radio systems group.

Nuno Souto graduated in aerospace engineering - avionics branch, in 2000 in Instituto Superior Técnico, Lisbon, Portugal and received his Ph.D. in 2006 . From November 2000 to January 2002 he worked as a researcher in the field of automatic speech recognition for Instituto de Engenharia e Sistemas de Computadores, Lisbon Portugal. He joined the ISCTE-Lisbon University Institute, as an assistant professor in 2006. He is a researcher at IT (Instituto de Telecomunicações), Portugal, since 2002 and has been involved in several international research projects and many national projects. His research interests include wireless networks, signal processing for communications, OFDM, single carrier transmission with frequency domain equalization, channel coding, modulation, channel estimation, synchronization, MIMO schemes, wireless sensor networks and unmanned aerial vehicles. He is a member of the IEEE Signal Processing Society.