The Journal of Global Positioning Systems

ORIGINAL ARTICLE Open Access

# Development of a GPS spoofing apparatus to attack a DJI Matrice 100 Quadcopter

Eric Horton and Prakash Ranganathan[*]

## Abstract

Global Positioning System (GPS) Spoofing attacks threaten technologies that our modern society depends on. To successfully develop defensive mechanisms against these attacks, methods to model the attacks and subsequently distinguish them from normal GPS operation must be developed. This paper primarily details the step-by-step implementation of a low-cost GPS spoofing and high-level spoofing data collection apparatus to model a simplistic spoofing attack that could be implemented with limited resources. The spoofing apparatus developed has been used to successfully attack a DJI Matrice 100 quadcopter and a portion of the collected spoofing data is presented.

**Keywords:** Global positioning system (GPS), Spoofing, Quadcopter, UAV

## Introduction

Global Positioning System (GPS) spoofing has become a well-known threat capable of crippling technologies that we rely on for day-to-day activities, impacting human lives and beyond. This necessitate the development of a comprehensive detection mechanism. This report outlines the step-by-step development of an apparatus to conduct a GPS spoof of a DJI Matrice 100 quadcopter and collect subsequent spoofing data. To provide a general understanding of the concept of GPS spoofing, the report will open with sections discussing the both the basics of GPS, spoofing, and state-of-the-art literature review carried in this area. The summary of the apparatus development will then cover the set-up of data-collection; monitoring software on the Matrice 100 (M100); and integration of hardware and software elements of the GPS spoofing apparatus. The results obtained for the GPS spoofing apparatus when used to attack both a cellular phone and the Matrice quadcopter are depicted. Finally, a simple comparison of the GPS data and peripheral sensor readings on the M100 quadcopter are studied along with a discussion of other methods to detect hostile spoofing attacks.

## Background and state of the art

### GPS overview

The Global Positioning System (GPS) relies on a constellation of satellites continually broadcasting data about each of their positions. These broadcasts from each satellite are conducted at different frequencies and modulation schemes depending on the application. The civilian GPS frequency band operates at 1575.42 MHz and is known as the L1-band.

The L1 civilian band is the primary focus of this paper. Each L1 signal broadcast by a GPS satellite is composed of the Navigation Message modulated on top of the Course Acquisition (C/A) code. The Navigation message itself details the ephemerides data (i.e., *orbital data*) for the satellite. The C/A code represents a pseudo random number (PRN) that is used by the GPS receiver to identify the satellite of origin.

A GPS receiver will shift the incoming L1 signals in time until a correlation peak is detected for a known PRN. During this period, the receiver will begin the process of retrieving the Navigation Message for that satellite. The time shift necessary to obtain the correlation peak is used to resolve the distance between the satellite and the GPS receiver. This process is continued in parallel for multiple and different PRNs, so the GPS receiver can obtain data for multiple different satellites. Because each PRN is designed to appear

[*] Correspondence: Prakash.ranganathan@engr.und.edu
University of North Dakota, Grand Forks, ND, USA

random, the correlator will only lock on the target satellite as having a correlation peak for that PRN (i.e. *the PRNs are orthogonal*).

Once at least four satellites have been identified via their PRNs using the C/A code; and enough ephemerides data has been obtained to calculate their position and time offset relative to the GPS receiver, the receiver will be able to calculate its own location relative to the satellites. At least four satellites are necessary to solve for three dimensions of position and the clock drift of the inaccurate receiver clock (4 unknowns, 4 equations), but usually many more are used for accurate positioning.

### GPS spoofing basics

GPS spoofing is accomplished by a system capable of mimicking the GPS signals associated with every satellite in the GPS constellation visible to the target receiver. The GPS transmission power of the fake GPS signals are higher than the real signals, resulting in the receiver locking onto them in favor of the true GPS. At this point the time shift of the fake signals can be manipulated to tamper with both the position and time reported by the receiver.

A sophisticated spoofer would gradually increase spoofing signal power at a time shift nearly equal to the current position of the receiver, thus allowing a seamless transition to the spoofed signal without any loss of lock or abrupt jumps in time or position. For a more comprehensive description of GPS spoofing, see (Tippenhauer et al., 2011).

### State-of-the-art background on GPS spoofing

Many research initiatives have recently implemented successful GPS spoofing attacks at varying degrees of sophistication. In 2015, a team of researchers from Mobile Security of Alibaba Group demonstrated the use of open source software and a software defined radio (SDR) to GPS spoof both a smartphone and smartwatch for time and position (Wang et al., 2015). The Unicorn Team of Qihoo 360 Technology Co. presented their development of an apparatus to both replay previously acquired GPS signals and generate custom spoofing Waveforms using Matlab and a software defined radio to spoof a smartphone, automobile, and DJI drone at DEF CON 23 (Huang & Yang, 2015). Two years later at DEFCON 25, Dave Karit of ZX Security demonstrated the use of a similar GPS spoofing setup to spoof an NTP server and manipulate the reported time (Karit, 2017). Aside from drones, personal smart devices, automobiles, and servers, researchers have also shown the susceptibility of phasor measurement unit (PMU) timing to GPS spoofing attacks, thus impacting power grid management software and human grid operators (Jiang et al., 2013; Shepard et al., 2012). In fact, one could conclude from these research examples that nearly any device relying on L1 civilian GPS is vulnerable to a GPS spoofing attack that can be implemented by combining relatively cheaper hardware and open source software.

In direct response to the threat of GPS spoofing attacks, researchers around the globe are developing novel methods of defense. The University of Calgary have compiled a detailed review with simulations of many anti-spoofing techniques in (Jafarnia-Jahromi et al., 2012) which will be a focal point for a later discussion in this paper. The authors in (Wang & Chakrabortty, 2016; Fan et al., 2017) propose algorithms capable of correcting timing for measurements from spoofed PMUs based on measurement trends within a large PMU network. A similar mechanism could potentially be applied to UAV swarms in communication with each other. The University of Ontario have successfully implemented a method that identifies spoofing attacks by the strong correlation between spoofed satellites signal parameters due to the nature of single transmitter spoofing apparatus inability to fake the multipath channel nature of a true GPS constellation (Li & Wang, 2016). A low-cost defense implementation based on free-running crystal oscillator comparison to received GPS signal timing has been proposed for Internet of Things (IoT) devices in (Arafin et al., 2017). This method would require no additional RF components or advanced signal processing techniques. Many UAV specific defensive methods, such as the threshold comparison of a UAV model estimation and GPS signal proposed in (Zou et al., 2016) exits.

## Methods

### DJI Matrice 100 setup and modifications

The overview of the setup of the DJI Matrice 100 (M100) includes an explanation of the interfacing with the DJI servers, the basic usage of the DJI On-Board software development kit (SDK), the integration of a ESP8266 Wi-Fi module for data collection during flight, and necessary modifications to the SDK. The setup being described in the following sections is depicted in Fig. 1.

#### *Interfacing with DJI*

DJI requires that their quadcopters are in constant communication with their servers during any flight operation. This includes both remote controlled flight using a UAV pilot or autonomous flight using software developed with the DJI On-Board SDK (OSDK). To ensure that this communication is established, any commands sent to the flight computer must be encrypted with an approved application ID that matches an ID logged in the DJI server. The Matrice 100 will verify

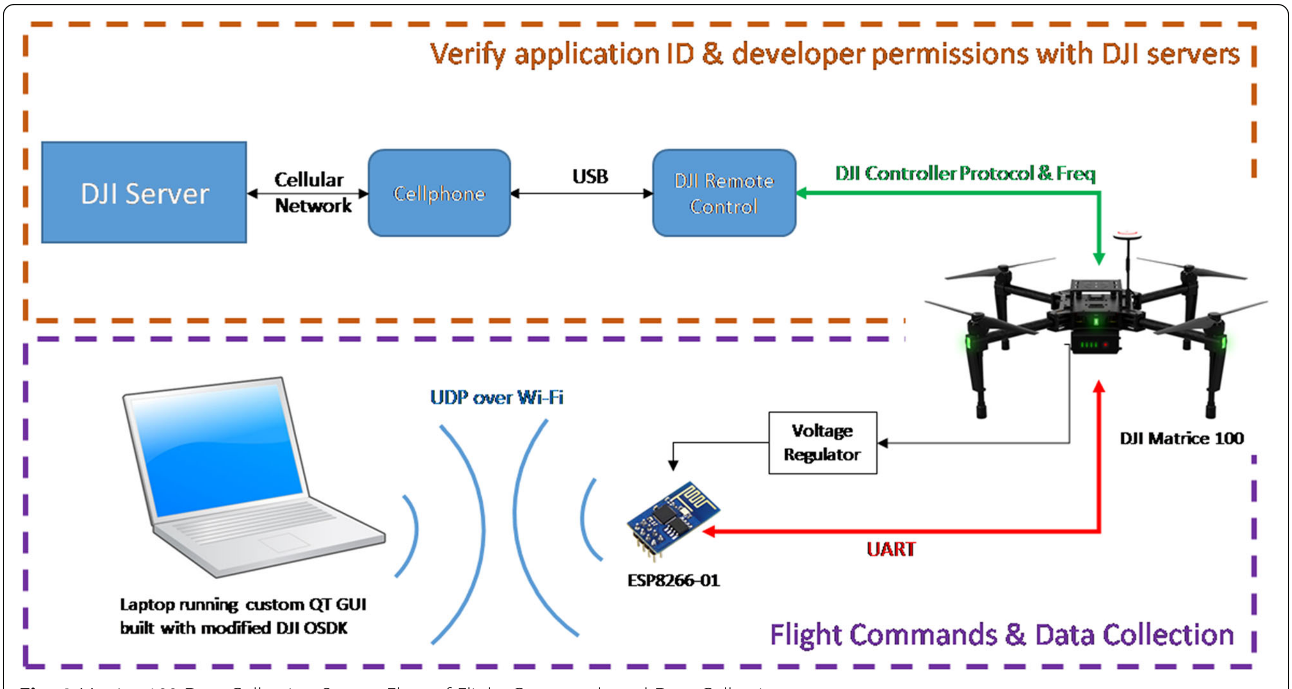

Verify application ID & developer permissions with DJI servers
DJI Server
Cellular Network
Cellphone
USB
DJI Remote Control
DJI Controller Protocol & Freq
DJI Matrice 100
Voltage Regulator
UDP over Wi-Fi
ESP8266-01
UART
Laptop running custom QT GUI built with modified DJI OSDK
Flight Commands & Data Collection

**Fig. 1** Matrice 100 Data Collection Set up. Flow of Flight Commands and Data Collection

this by connected to the server through the DJI remote controller and then through the cellular network via a cellphone running the DJI Go application. Obtaining an application ID along with downloading the OSDK code is facilitated through the DJI developer website (DJI, 2017a).

### *Using the on-board SDK*

The On-Board SDK includes a set of C++ classes that establish UART communication with the DJI Matrice 100 as well as facilitate a variety of flight commands and data requests. By creating code with the OSDK to run on an embedded processor, the processor can be placed on-board the DJI Matrice 100 during flight enabling autonomous capabilities. To facilitate rapid development of the on-board software as well as ease of data collection, a transparent Wi-Fi to UART bridge was create using an ESP8266 Wi-Fi module along with modifications to the SDK. This allowed for all software to be developed and run on a laptop computer with command and response to and from the Matrice 100 facilitated using the Wi-Fi-UART link.A custom GUI was created using the modified OSDK and QT development environment to collect data during the GPS spoofing tests.

### *OSDK modifications and ESP8266 integration*

As stated previously, the UART communication supported by the OSDK was abandoned in favor of a Wi-Fi connection using a laptop computer. This modification required changes to the OSDK *DJI_Pro_HW.cpp* file as listed in Table 1.

An ESP8266 was then used on the Martice 100 to translate the UDP packets coming over the Wi-Fi connection to a simple UART stream at the required baud rate. The ESP8266 was flashed with a transparent Wi-Fi UDP to UART binary known as esp-link. Details on how to flash the ESP8266 with esp-link along with the precompiled binaries themselves can be found at the esp-link github (JeeLabs, 2017) Because the ESP8266 requires 3.3 V input voltage to function it was connected to the Matrice 100 battery through a 5 V buck

**Table 1** Overview of DJI_Pro_HW.cpp modifications

| Member Function | UART | Wi-Fi |
|---|---|---|
| Pro_HW_Create_Instance() | create serial object | Create UDP object |
| run() | Loop read data from serial port | Create UDP Socket, loop Pro_HW_recv() |
| Pro_HW_recv() | N/A | Receive UDP Packet data |
| Pro_HW_send() | send data over serial port | Write data to UDP socket |
| Other functions/members | Replace serial port with udp socket object | |

regulator and properly configured linear regulator that were readily available.

### *Laptop QT graphical user Interface*

A simple graphical user interface (GUI) was created using the QT development environment along with the modified DJI OSDK. This GUI, Fig. 2, was implemented by building off an example implementation from the DJI OSDK GitHub repository (DJI, 2017b).

The custom GUI includes an interface to change the hovering altitude of the M100 as well as to fly in either the x or y directions relative to the M100. This was accomplished by sending various flight commands created in the DJI OSDK over the Wi-Fi to UART data link to the flight computer. The GUI also includes a display of GPS data including the latitude, longitude, altitude, and health. In addition to the GUI interface, while running, the QT program is constantly logging all accelerometer, gyroscope, quaternion, and gps data from the M100 status messages in a. CSV file for post-processing.

## GPS spoofing apparatus

The working spoofing apparatus used at the University of North Dakota encompasses a custom hardware setup running open source GPS simulation software. The discussion of the setup will be broken down into an overview of its hardware and software components. After discussing these components, a typical GPS spoofing session will be described including steps taken and commands utilized. It is important to note that the current working spoofing apparatus at the UND is only capable of spoofing using pre-generated I-Q signal representation. The spoofed signal also does not attempt to reduce large jumps in signal power, position, or GPS time. Therefore, the spoofing apparatus represents a proof of concept in its implementation to attack Matrice 100 UAS (see Fig. 3 for Matrice 100 specifications).

### *Basic spoofing hardware*

The hardware necessary to spoof the M100 include a computer capable of generating the spoofed signal's IQ data stream; a software defined radio (SDR) that will transform the IQ data into RF output; an antenna that operates at 1575.42 MHz frequency used by the L1 GPS signal; and an appropriately selected attenuator to ensure that the spoofed signals do not travel beyond the testing radius. The spoofing setup at the University of North Dakota (UND) uses a low-cost Dell laptop with Windows 10 operating system connected to a BladeRFx40 SDR with a 50 dB attenuator connecting to a Garmin GPS antenna on one of its transmit SMA connectors. This hardware setup and the bladeRF I-Q modulation is shown in Figs. 4 and 5 respectively.

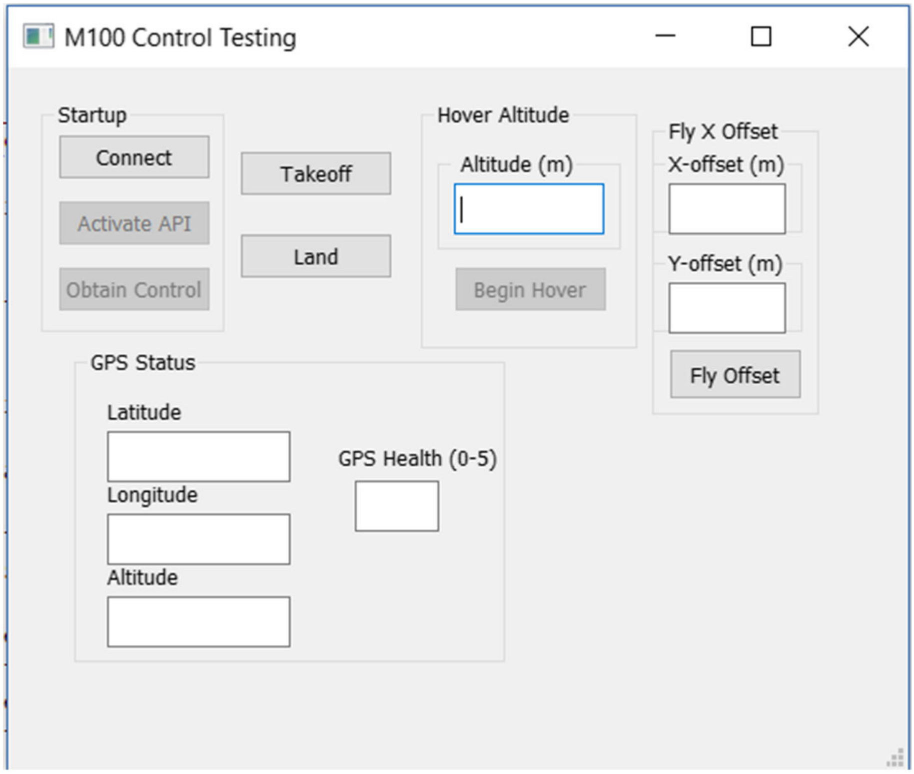
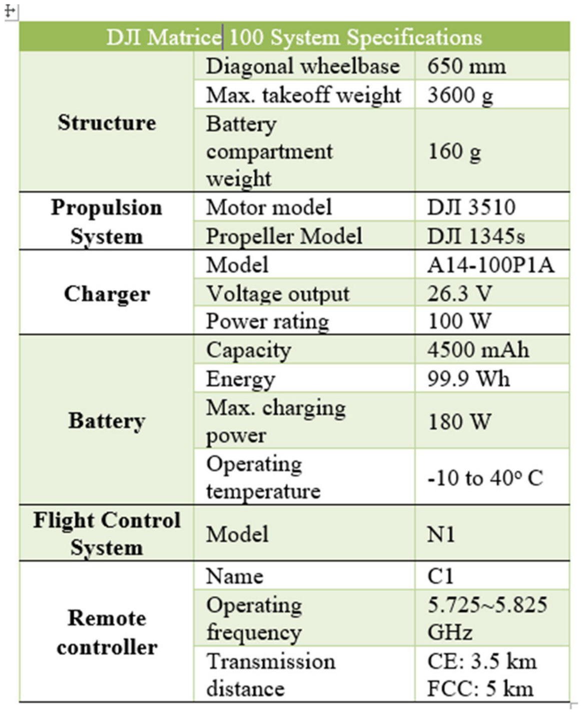


**Fig. 2** GUI Interface for Matrice 100. Real Time Control Test Environment that show GPS status and flight parameters



| DJI Matrice 100 System Specifications | | |
|---|---|---|
| Structure | Diagonal wheelbase | 650 mm |
| | Max. takeoff weight | 3600 g |
| | Battery compartment weight | 160 g |
| Propulsion System | Motor model | DJI 3510 |
| | Propeller Model | DJI 1345s |
| Charger | Model | A14-100P1A |
| | Voltage output | 26.3 V |
| | Power rating | 100 W |
| Battery | Capacity | 4500 mAh |
| | Energy | 99.9 Wh |
| | Max. charging power | 180 W |
| | Operating temperature | -10 to 40° C |
| Flight Control System | Model | N1 |
| Remote controller | Name | C1 |
| | Operating frequency | 5.725~5.825 GHz |
| | Transmission distance | CE: 3.5 km<br>FCC: 5 km |

**Fig. 3** Matrice 100 Specifications. Specifications indicating battery information, flight and remote controller information

### *Spoofing hardware range*

An approximate range of the GPS spoofing setup described can be calculated by combining the known parameters of the GPS L1 signal power at the Earth's surface, carrier frequency, the transmit power of the bladeRF, and the losses due to the software settings and attenuator.

By manipulating the Friis Transmission Equation:

$$P_r = P_t + G_t + G_r - FSPL - \sum Loss \tag{1}$$

$$FSPL = 20\,\log\left(\frac{4\pi}{c}\right) + 20\,\log(f) + 20\,\log(d) \tag{2}$$

$$d = 10^{\left(-(P_r - G_r) + P_t + G_t - 20\,\log\left(\frac{4\pi}{c}\right) - 20\,\log(f) - \sum Loss\right)/20} \tag{3}$$

The values to be used in eq. 3 to calculate an estimated range of the GPS spoofing apparatus described are shown in Table 2. Table 3 shows the results for the nominal range calculation as well as when either the attenuator or output power settings of the bladeRF are not properly selected.

The results of Table 3 demonstrate the importance of the attenuator and bladeRF gain settings in ensure that the spoofing apparatus range is properly controlled.

Note that loss parameter used in eq. 3 to derive the approximate spoofing range did not account for the attenuation due to cables, atmosphere, or physical obstacles. As such the simple calculation utilized represent an order-of-magnitude estimate for line of site range.

### *Basic spoofing software*

The software used to generate the spoofed GPS signal's IQ data stream that will be fed the BladeRFx40 SDR is the open-source GPS-SDR-SIM created by Takuji Ebinuma which can be found on GitHub at: *https://github.com/osqzss/gps-sdr-sim (Ebinuma, 2017)*. The GPS-SDR-SIM software can generate an IQ data stream for both static location and dynamic user-defined motion profiles. The IQ data is generated using a RINEX navigation file for GPS

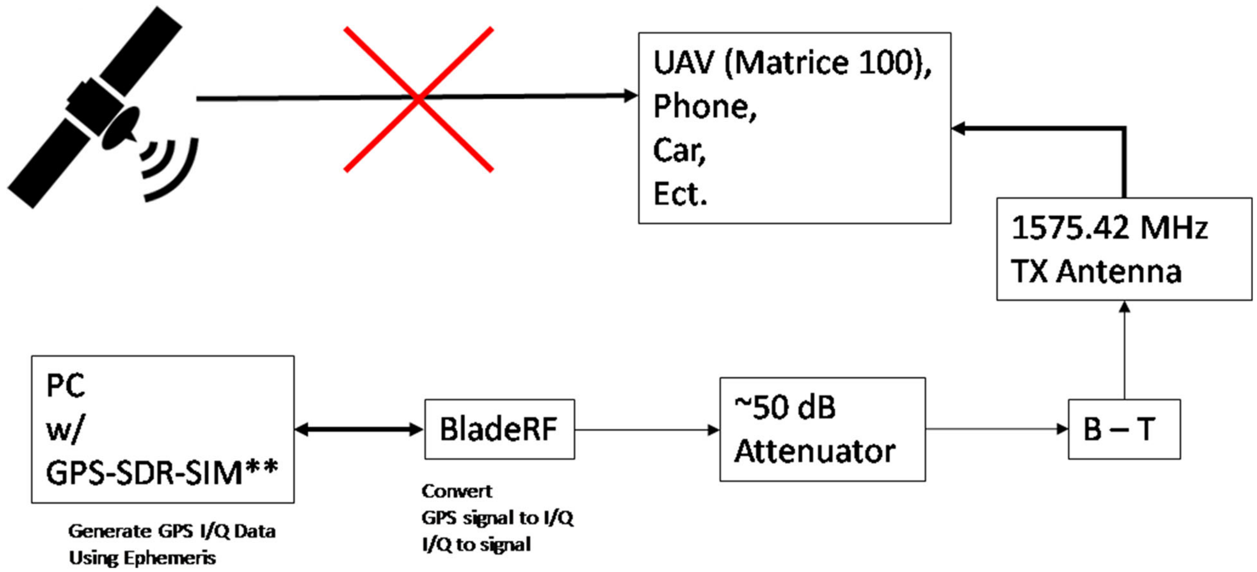


**Fig. 4** GPS Spoofing Apparatus. Test Bed Framework showing SDR and UAS integration

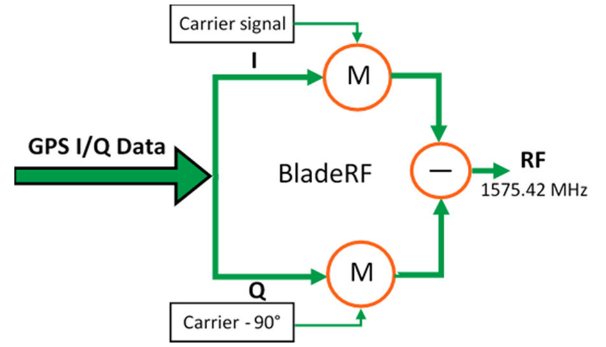


**Fig. 5** GPS I-Q Modulation. Carrier Modulation with IQ Data outputting RF frequency of 1575.42 MHz

ephemerides for the intended time of the spoofed signal (Ebinuma, 2017). Once built on a computer the software has a command line interface with a variety of possible arguments that allow the user to generate the IQ data stream of a given duration from the RINEX file and either a static earth-centric location or dynamic location file. More details on these options as well as the inner workings of the software can be found at the previously mentioned GitHub location (Ebinuma, 2017).

### *Spoofing attack setup*

Once all necessary hardware have been connected as per Fig. 4, and the laptop computer has built both the BladeRFx40 command line interface and GPS-SDR-SIM a spoofing attack can be run. The low gain settings to be used on the BladeRFx40 and the 50 dB attenuator will reduce signal strength, but the spoofed signal will still be stronger than real GPS within an approximately 25 m range (as identified previously), therefore the spoofing attack should be undertaken in a semi-shielded environment to ensure that spoofed GPS signals do not interfere with outside devices. At the University of North Dakota, the signals and systems laboratory has been typically utilized for this purpose.

If the GPS-SDR-SIM has been added to the system path, the IQ data stream file can be generated for a static spoof using the command (Ebinuma, 2017):

*gps-sdr-sim -e < RINEX Ephemeris file > –l < lat,long,alt > –d < duration>.*

**Table 2** Friis equation parameters

| Symbol | Value | Description |
|---|---|---|
| ($P_r$ - $G_r$) | – 130 dBm | Minimum GPS power level at Earth's Surface |
| $P_t$ | 6 dBm | BladeRF nominal TX power |
| $G_t$ | 3.5 dBi | Transmit antenna gain |
| f | 1575.42 MHz | GPS L1 carrier frequency |
| $Loss_1$ | 50 dB | 50 dB attenuator |
| $Loss_2$ | 25 dB | BladeRF gain settings |

**Table 3** Approximate spoofing range

| Condition | Spoofer Approximate Range (d) |
|---|---|
| Nominal | 25.4 m |
| No bladeRF settings | 452 m |
| No 50 dB attenuator | 8046 m |

For example, creating a 300 s GPS signal corresponding to Grand, Forks North Dakota for April 11, 2013 at 12:00 AM would correspond to the command:

*gps-sdr-sim -e brdc1010.13n -l 47.9253, 97.0329, 0 -d 300.*

Note that the "brdc" RINEX ephemeris files can be located at *ftp://cddis.gsfc.nasa.gov/gnss/data/daily/* under the year then day of the year (1 through 365) and then XX.n directory (where XX is the last two digits of the year). The brdc file is downloaded and then unzipped it to the directory that the command will be run from. Dynamic location streams can also be generated per the documentation found at the previously mentioned GitHub.Once the IQ data stream is generated, it will be saved by default in gpssim.bin. The data can then be piped to the BladeRF with the default GPS-SDR-SIM settings by using the command:

*bladeRF-cli -s bladerf.script.*

This script corresponds to the setting the frequency to 1575.42 MHz, sample rate to 2.6 MHz, bandwidth to 2.5 MHz, transmit gain to -25 dB, calibrating the transmit output, selecting gpssim.bin as the IQ data file, and finally starting the transmission. The entire command line process listed in this section is also well document in both (Wang et al., 2015; Ebinuma, 2017).

## Results

### GPS spoofing results

Successful GPS spoofing attacks were undertaken on both an Android smartphone and then the DJI Matrice 100 (M100) quadcopter. This section will depict the results of these attacks.

### *Android smartphone spoof results*

Before testing on with the M100 a static location spoof was verified using a HTC Desire 626 s cellphone with the "GPS Data – location status fix" application running for diagnostics.

To quickly spoof the cellphone, both mobile data and Wi-Fi were turned off to prevent the device from receiving location data from any external network. The GPS spoofing apparatus was run as described previously using a static location spoof first near Shanghai, China and then in middle of the Pacific Ocean. The Google Maps output from the phone for both attacks are shown in Fig. 6.

It is also apparent from the output of each test that the timing was also successfully spoofed as both tests

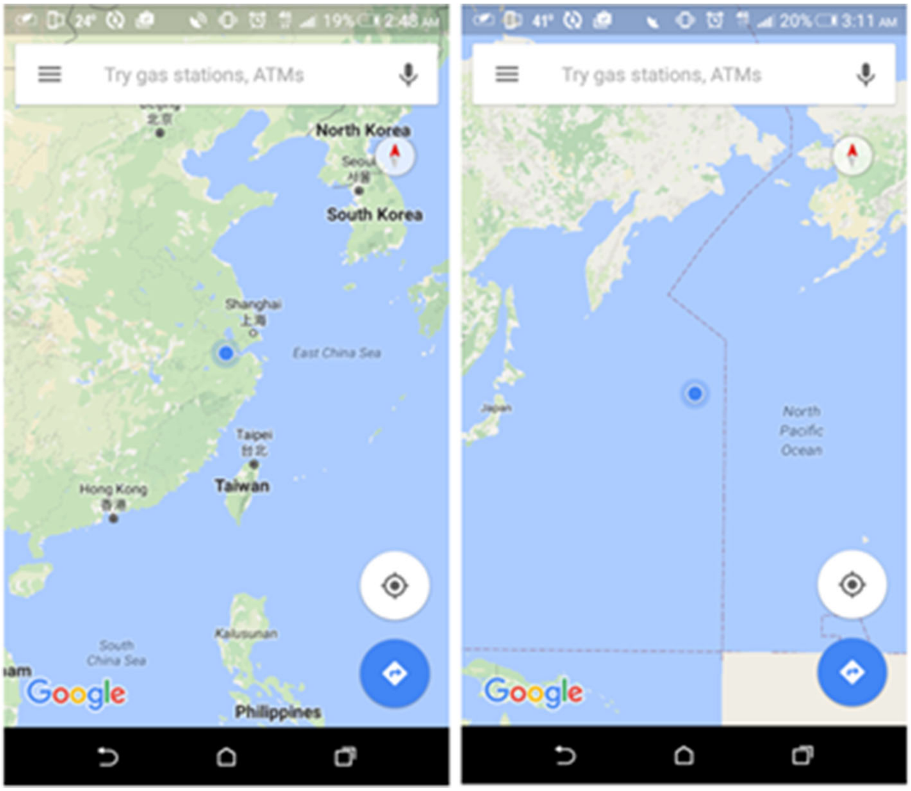


**Fig. 6** Cellphone GPS spoof tests. Fake locational and Actual positions

were run in Grand Forks, ND at approximately 6 PM. Figure 7 show the output of the GPS Data application's satellite view compared to the satellite signals being generated by the spoofing software shown in the command line. The application clearly shows that satellites 7, 6, 7, 13, 15, 20, 29, and 30 being used for the GPS lock with high signal strengths. It can also be seen that these are precisely the satellites being spoofed by GPS-SDR-SIM. Note that satellite 12 is being spoofed, but not used by the cellphone for gps lock.

### *DJI Matrice 100 results*

The M100 was first subject to a static location spoof in the same manner as the Android smartphone. The M100 was spoofed much more quickly than the smartphone most likely because of the strength of its GPS receiver along with the fact that it uses GPS alone for positioning rather than cellular networks and Wi-Fi. The DJI Go application displayed the location of the M100 to be in Australia precisely as configured in the spoofing setup. Furthermore, the latitude and longitude data obtained via the Wi-Fi to UART interface with the OSDK and logged by the QT application showed that the M100 was reporting its location at the same location in Australia. Both the DJI Go application output and the location transmitted over the OSDK interface plugged into Google Maps are depicted in Fig.8. After conducting the static location spoof, a predefined dynamic location profile included as an example in the previously mentioned GPS-SDR-SIM GitHub repository was fed to the GPS-SDR-SIM software.

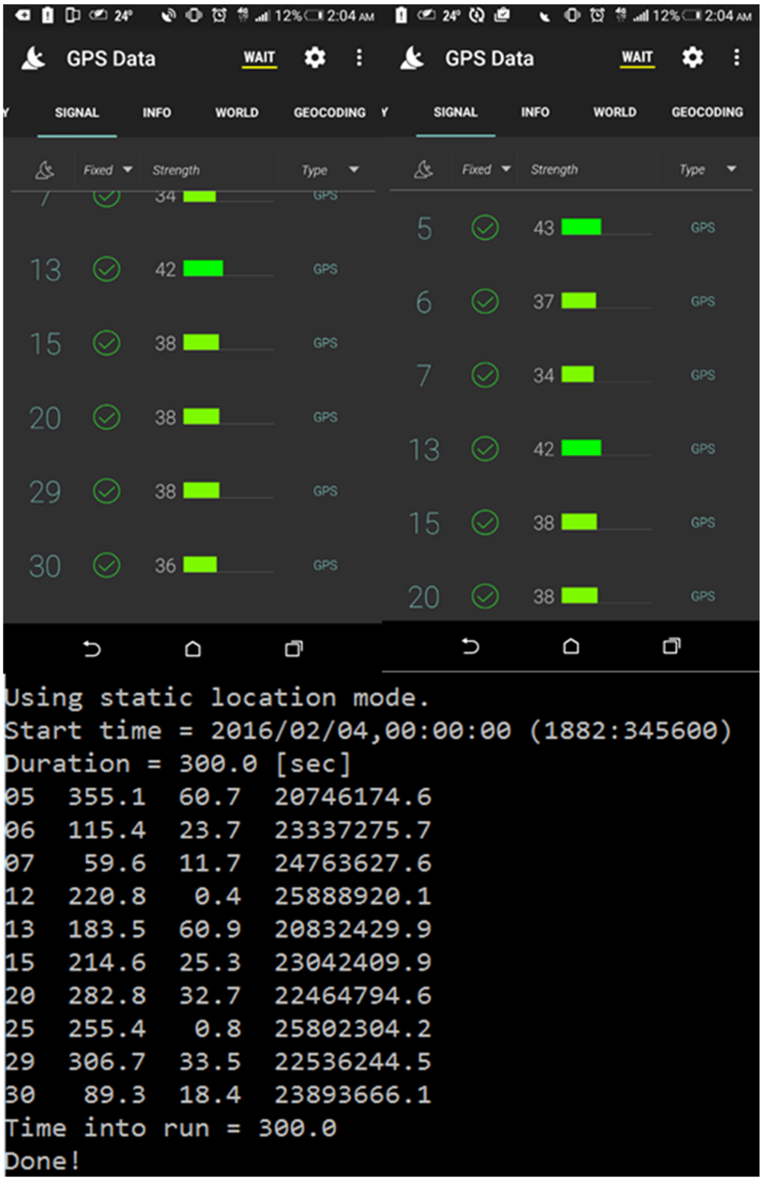
The output of the DJI Go application location for this test is shown in Fig. 9. Fusion sensor data was also collected using the DJI OSDK Wi-Fi interface. The DJI Matrice 100 was sitting idle on a test bench during the attack with its propellers removed to keep the drone from damaging itself. The DJI Go App reported the drone moving in a large circular path and the Matrice 100 motors were actuating at full power to counteract the spoofed movement. If the drone would have been hovering at this point it would surely have crashed attempting its location against the falsified movement profile.

# Discussion

## GPS spoofing detection and defenses

### *Basic sensor comparison spoofing detection*

The "fusion" sensor data, a preprocessed combination of the accelerometer and gyroscope data telemetered by the Matrice 100 that approximates the velocity along absolute three-dimensional axes (DJI, 2017b), versus the apparent



| | Fixed | Strength | Type |
|---|---|---|---|
| 7 | ✓ | 34 | GPS |
| 13 | ✓ | 42 | GPS |
| 15 | ✓ | 38 | GPS |
| 20 | ✓ | 38 | GPS |
| 29 | ✓ | 38 | GPS |
| 30 | ✓ | 36 | GPS |

| | Fixed | Strength | Type |
|---|---|---|---|
| 5 | ✓ | 43 | GPS |
| 6 | ✓ | 37 | GPS |
| 7 | ✓ | 34 | GPS |
| 13 | ✓ | 42 | GPS |
| 15 | ✓ | 38 | GPS |
| 20 | ✓ | 38 | GPS |

```
Using static location mode.
Start time = 2016/02/04,00:00:00 (1882:345600)
Duration = 300.0 [sec]
05  355.1  60.7  20746174.6
06  115.4  23.7  23337275.7
07   59.6  11.7  24763627.6
12  220.8   0.4  25888920.1
13  183.5  60.9  20832429.9
15  214.6  25.3  23042409.9
20  282.8  32.7  22464794.6
25  255.4   0.8  25802304.2
29  306.7  33.5  22536244.5
30   89.3  18.4  23893666.1
Time into run = 300.0
Done!
```

**Fig. 7** Cellphone GPS Statistics. Statistics showing GPS with SDR-SIM CLI output

velocity calculated using the GPS data both obtained during the dynamic movement spoof are shown in Fig. 10. As described previously the Matrice 100 was sitting idle during this test which is confirmed by the output sensor data's contrast to the GPS data.

Although this data represents the extreme case of the DJI drone have spoofed movement while no real movement exists, the data clearly shows the discrepancy between the GPS data and the other sensors aboard the Matrice 100. It follows that a simple fact checking of peripheral sensor data versus the GPS positioning could be used as a means of spoofing defense.

The inherent noise of the sensors poses a significant challenge to the implementation of a simple sensor fact check for GPS spoofing. A Kalman filter using the accelerometer and gyroscope data was implemented on the data to reduce any sensor noise, but further testing with this Kalman filter along with the introduction of additional sensors such as the cameras, or ultrasonic sensors currently implemented on the Matrice 100 to improve the Kalman filter accuracy remains future work.

### *Sophisticated spoofing detection methods overview*

Some detection methods for anti-spoofing, specifically within UAS applications, include:

i. Synthetic Array Spoofing Discrimination: The movement of a single receiver antenna over time

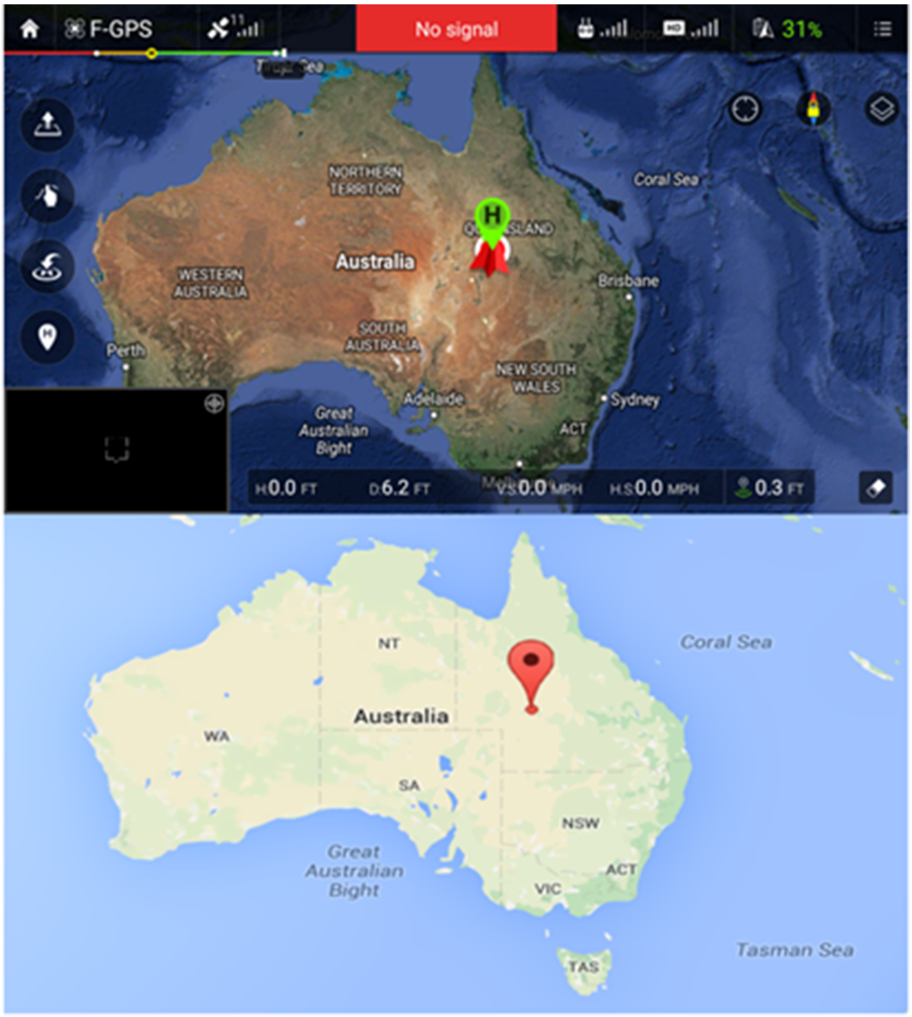


**Fig. 8** M100 DJI Go App Location. OSDK locational output

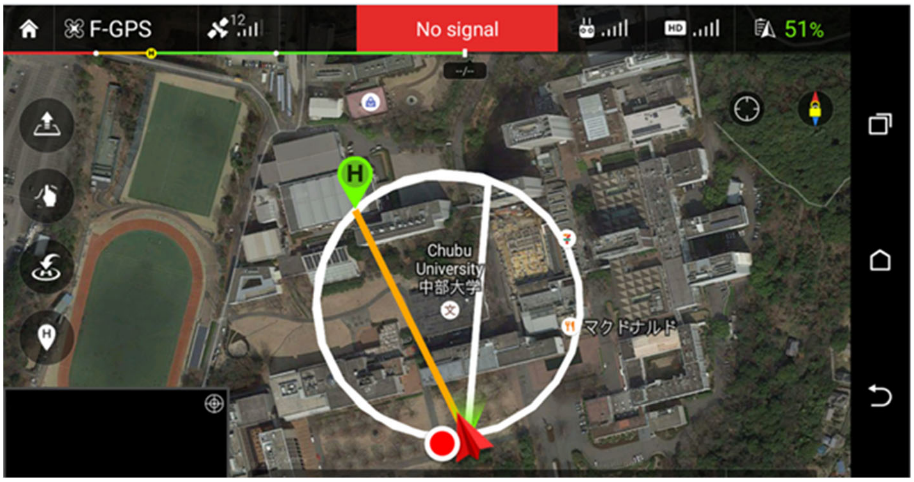


**Fig. 9** M100 dynamic location GPS spoofing. Perimeter Range

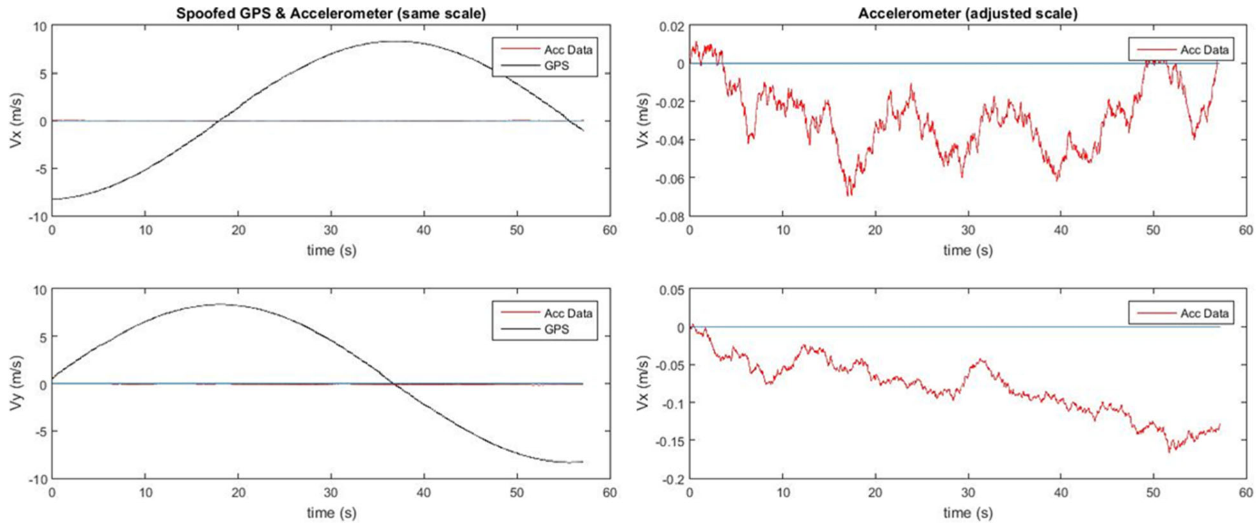


**Fig. 10** Calculated Velocites.tif. Calculated Velocites

can be used as a type of synthetic array for spoofing detection by monitoring the correlation in the changing amplitude and phase angle for the GPS signals. For a spoofed GPS signal, the individual satellite will exhibit strong correlation changes with antenna movement, which will not occur for true GPS received from multiple separate satellite transmitters. This method shows promise for a continually moving UAV.

ii. Power Variation vs Receiver Movement: This method of spoofing detection is similar to the synthetic movement based array except that it instead monitors only the change in power associated with movement of the receiver. As the receiver moves all signals originating from a single spoofer transmitter will increase or diminish in power unlike a true GPS constellation. This method requires accurate power monitoring of the very low power GPS signals and relies on enough UAV movement to effect power at a detectible level.

iii. Consistency Check with Other Navigation and Position Technologies: This method relies on comparing other sensors and position estimates with the GPS position. This is the primary focus of UAV GPS spoofing detection related work conducted at the University of North Dakota thus far due to the availability of gyroscopes, accelerometers and even cameras readily available on most commercial UAVs.

A more comprehensive review of GPS spoofing detection methods, including those listed here, has been conducted in (Jafarnia-Jahromi et al., 2012).

### *GPS spoofing defense*

Once a spoofed GPS signal has been detected, the next significant challenge is to take some sort of corrective action whether it be a UAV relying on other means of navigation via sensor networks or a smartphone obtaining positioning via the cellular network. These methods of defense remain a significant research opportunity and a point of future work for this research.

## Conclusion

A GPS spoofing apparatus has been developed at the University of North Dakota (UND) to implement spoofing attacks for the use of further development of detection and defensive mechanisms. The steps undertaken to implement this apparatus were detailed as well as the modifications to the Matrice 100 quadcopter necessary for effective spoofing attack data collection. The results of the spoofer in action have been presented along with a showcase of some of the data available from the Matrice 100 during operation.

**Acknowledgements**
This research work is sponsored by Rockwell Collins Corporation, Cedar Rapids, IA, and authors appreciate the support and effort from the company, and specifically Roger French, Principal Engineer at Rockwell Collins.

**Funding**
Rockwell Collins Corporation, AWARD Number UND22526.

**Data statement**
Data sharing not applicable to this article as no datasets were generated or analyzed during the current study. If you do not wish to publicly share your data, please write: "Please contact author for data requests."

**Authors' contributions**
EH carried out the Kalman filter design, developed coding on GPS spoofing, and implementation. PR developed the software defined radio based configuration, and conceived of the study, and participated in its design and

coordination and helped to draft the manuscript. Both authors read and approved the final manuscript.

### Competing interests

The authors declare that they have no competing interests.

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 22 November 2017 Accepted: 13 June 2018
Published online: 03 July 2018

### References

Arafin M.T., Anand D. and Qu G., "A Low-Cost GPS Spoofing Detector Design for Internet of Things (IoT) Applications", 2017, pp. 161–166

DJI, "DJI Developer Website." 2017a. Internet: https://www.dji.com/

DJI, "dji-sdk/Onboard-SDK (GitHub repository)." Internet: https://github.com/dji-sdk/Onboard-SDK, 2017b

Ebinuma T., "GPS-SDR-SIM (GitHub repository)." Internet: https://github.com/osqzss/gps-sdr-sim, 2017

Fan X, Du L, Duan D (2017) Synchrophasor data correction under GPS spoofing attack: a state estimation based approach. IEEE Transactions on Smart Grid, vol PP:1–1

Huang L. and Yang Q., "GPS spoofing low-cost GPS simulator," 2015

Jafarnia-Jahromi A, Broumandan A, Nielsen J, Lachapelle G (2012) GPS vulnerability to spoofing threats and a review of Antispoofing techniques. Int J Navigation and Observation 2012(127072):16. https://doi.org/10.1155/2012/127072

JeeLabs, "esp-link (GitHub repository)." Internet: https://github.com/jeelabs/esp-link, 2017

Jiang X, Zhang J, Harding BJ, Makela JJ, Dominguez-Garcia AD (2013) Spoofing GPS receiver clock offset of Phasor measurement units. IEEE Trans Power Syst 28:3253–3262

Karit D., "Using GPS spoofing to control time," 2017

Li H. and Wang X., Detection of GPS spoofing through signal multipath signature analysis, pp 1–5, 2016

Shepard DP, Humphreys TE, Fansler AA (2012) Evaluation of the vulnerability of phasor measurement units to GPS spoofing attacks. Int J Crit Infrastructure Prot, vol 5 12(/01):146–153

Tippenhauer N.O., Popper C., Rasmussen K. and Capkun S., On the requirements for successful GPS spoofing attacks, 2011, pp. 75–86

Wang K., Chen S. and Pan A., "Time and position spoofing with open source projects," 2015

Wang Y. and Chakrabortty A., Distributed monitoring of wide-area oscillations in the presence of GPS spoofing attacks, pp. 1–5, 2016

Zou Q, Huang S, Lin F, Cong M (2016) Detection of GPS spoofing based on UAV model estimation. pp.:6097–6102