12-31-2018

Forensic Analysis of a Crash-Damaged Cheerson CX-20 Auto Pathfinder Drone

Ian N. McAteer
School of Science, Edith Cowan University, Perth, Australia, imcateer@westnet.com.au

Peter Hannay
School of Science, Edith Cowan University and Asterisk Information Security, Perth, Australia, p.hannay@ecu.edu.au

Muhammad I. Malik
School of Science, Edith Cowan University, Perth, Australia, mimalik@our.ecu.edu.au

Zubair Baig
School of Information Technology, Deakin University, Geelong, Australia, z.baig@ecu.edu.au

Follow this and additional works at: https://commons.erau.edu/jdfsl
Part of the Computer Law Commons, and the Information Security Commons

Recommended Citation
Available at: https://commons.erau.edu/jdfsl/vol13/iss4/5

This Article is brought to you for free and open access by the Journals at Scholarly Commons. It has been accepted for inclusion in Journal of Digital Forensics, Security and Law by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu, wolfe.309@erau.edu.
FORENSIC ANALYSIS OF A CRASH-DAMAGED CHEERSON CX-20 AUTO PATHFINDER DRONE

Ian Noel McAteer¹, Peter Hannay¹,², Muhammad Imran Malik¹, Zubair Baig³

¹School of Science, Edith Cowan University, Perth, Australia
²Asterisk Information Security, Perth, Australia
³School of Information Technology, Deakin University, Geelong, Australia
imcateer@our.ecu.edu.au, p.hannay@ecu.edu.au, peter.hannay@asteriskinfosec.com.au, mimalik@our.ecu.edu.au, zubair.baig@deakin.edu.au

ABSTRACT

Long gone are the days when Unmanned Aerial Vehicles (UAVs) and drones (multirotor UAVs) were the exclusive domain of the military for surveillance or tactical strike purposes. For relatively little money mainly due to high-tech progression in microprocessor design, anyone can now purchase a drone with GNSS-tracking capabilities and can support a live high-resolution video feed to its flight controller. The global population of drones has skyrocketed in recent years as this new technology has been embraced for both its recreational and commercial applications. However, the more nefarious members of society have also recognized the potential for using drones to partake in criminal and terrorist activities. In such cases, the relatively inexpensive drones are often sacrificed to facilitate a quick escape once the criminal act has been completed. The pioneering field of UAV/drone forensics has grown out of the challenge law enforcement faces in examining the abandoned hardware for digital traces that can be used to identify the criminals themselves.

Keywords: Drone Forensics, UAV Forensics, Global Positioning System, Global Navigation Satellite System (GNSS), GNSS Forensics.

1. INTRODUCTION

UAVs are much more than a model aircraft piloted via radio transmissions from a flight controller. They are part of an entire Network of Things (NoT) ecosystem of sensors (satellite positioning, accelerometers, gyros, compass, barometer, high-resolution digital camera, onboard diagnostics); controllers (pilot handsets, smartphones); data storage (firmware, onboard memory, apps) and the GNSS satellites themselves in orbit (Luo, 2016; Marks, 2017) and falls under the category of smart mobility within the smart city model (Baig et al., 2017). Baig et al. (2017) further argues that with each drone flight, digital traces are left in the form of onboard log files (flight logs, connection logs, and system logs) retained in memory, GNSS data or in photographic metadata, all of which the forensic analyst can use to investigate the drone, and thus its pilot’s activities.
In September 2017 the opportunity arose to conduct a research-based forensic investigation of a crash-damaged Cheerson CX-20 Auto-Pathfinder drone. The drone’s owner is a recreational drone enthusiast. The drone itself had crashed beyond repair in early Jan 2015, and since then it had lain untouched in the owner’s home.

Figure 1. Crash-Damaged Cheerson CX-20 Auto-Pathfinder as Received from its Owner.

2. BACKGROUND

The use of Unmanned Aerial Vehicles (UAVs) can perhaps be traced back as far as 1849 when Austria used bomb-equipped balloons to attack Venice (Naughton, 2007). More recent technological advancements in the UAV field, not surprisingly, came from the military’s desire to preserve pilots’ lives (Wagner, 1982). Through successive military conflicts since the 1960s, drones have played an increasingly important role in reconnaissance and surveillance operations.

Since the turn of the century, military UAVs have developed from a passive reconnaissance role to becoming armed (Strike-Enabled UAVs) and capable of launching missiles. In the early 2000s, the MQ-1 Predator drone’s payload capacity restricted armaments to two AGM-114 laser-guided Hellfire missiles (Rowley, 2017). Currently, larger drones such as the MQ-9 Reaper (‘Predator B’) and Avenger (‘Predator C’) can carry maximum payloads of 1747 kg (General Atomics Aeronautical, 2015a) and 2948 kg (General Atomics Aeronautical, 2015b) respectively.

The technological advancements in digital electronics, GNSS positioning and portable power supplies in recent years has also spawned a considerable development of drones for civil and recreational purposes. All manner of aerial reconnaissance can be achieved using UAVs at a considerably reduced cost compared to using helicopters or aircraft, and without putting a pilot’s life at potential risk.

However, criminal elements have also realized that they can exploit a drone’s capabilities to attain their own goals. Drones are frequently used to fly contraband into prisons throughout the world (Stitt, 2017). They have been used to violate another nation’s airspace (Associated Press, 2013). They have been used to incite civil unrest (Ames & Ibrulj, 2014) and to make political protests (Associated Press, 2015). Terrorists are also now taking advantage of drone characteristics too. In 2017, ISIS (Islamic State in Syria) drones armed with grenade-sized munitions were observed attacking US-backed forces in the battle for Raqqa (France24.com, 2017). Imagine the devastation that could be caused by a drone dispersing Sarin gas upwind from Times Square in New York at midnight on New Year’s Eve. Most recently extensive flight disruption was caused at Gatwick Airport when a drone entered the restricted air space around the airport (bbc.com, 2018). Whether the motivation behind this latest incident was a mere prank or with future terrorist potential in mind, the ensuing investigation and the difficulties in finding the culprits highlights the need to further research in this new field of drone forensics and develop worldwide standards accordingly.

The fact that commercial and recreational drones are relatively cheap is an advantage
for the forensic investigator. Compared to the objective a criminal element intends to accomplish, the cost of the drone is her/his lowest priority. As such, the drones employed for such purposes are most likely to be abandoned immediately after the intended action has been taken. Drone forensic analysis, therefore, is likely to become an ever more prominent tool in law enforcement’s arsenal to try to bring such criminals to justice.

3. METHOD
The forensic analysis undertaken during this project followed the OSCAR principle (Davidoff & Ham, 2012), which consists of the following five steps:

1. Obtain Information
2. Strategy
3. Collection of Evidence
4. Analysis
5. Reporting

3.1 Obtain Information
The first step of the investigation was to extract as much information relating to the Cheerson CX-20 Auto Pathfinder Drone from the Internet as possible. Documentation found included technical specifications (IMG Electronics Co. Ltd., n.d.), owner/user operation manuals (Cheerson, 2015; Naudin, 2014), and wiring diagrams (www.cita.asia, n.d.).

3.2 Strategy
The following step-by-step plan was envisioned for the extraction of any residual data from the damaged drone¹, and to determine how accurately the drone positioned itself according to received GNSS signals. However, the success of this investigation would ultimately depend on the degree of damage that had been sustained at the electrical and mechanical component level:

3.2.1 Extraction of Residual Flight Data
- Inspect the internal components of the crash-damaged CX-20 to confirm the hardware specifications published by the manufacturer and determine if any wiring needed to be replaced or inserted.
- Establish if the CX-20 powers up using a compatible power supply.
- Remove the ArduPilotMega (APM) CPU circuit board.
- Confirm if serial communications can be established between the APM CPU board and a computer via a micro USB cable.
- Install manufacturer-recommended Mission Planner software. As this is Windows-only software, it was installed on a desktop PC running Windows-10.
- Connect the APM CPU board on the Mission Planner software.
- Extract any available log files representing residual flight data before the CX-20’s crash.
- Analyze the residual flight data.

¹The condition of the drone can be seen in Figure 1. The upper cowling, including the GPS antenna mounting, were entirely missing, as where the propellers. The lower cowling was cracked in several places. Electronic components appeared physically intact, but with several wires disconnected.
3.2.2 Generation of Simulated Flight Data

Figure 2 shows the configuration of the hardware setup housed in a Faraday cage\(^2\) to be used to conduct GNSS simulation tests.

Figure 2. GNSS Simulation Configuration within a Faraday Cage.

Hannay (2017) uses the terms ‘open sky’, referring to the outdoor real-world surroundings, as opposed to the ‘closed sky’ environment within the confines of a Faraday cage. The latter improves repeatability by removing the random effects of “atmospheric conditions, unpredictable multipath effects, and infrastructure change during the research period”.

Preparations were made to conduct the following experiments:

1. Compare the signals received from a simulated flight path as recorded by monitor software and the drone control software (Test A).

2. Compare the signals received from a residual flight path recovered from the drone as recorded by monitor software to the data existing in the drone control software (Test B).

The purpose of these experiments is to test the efficacy of forensic procedures when applied to data acquired from crashed drones. This can only be possible through a rigorous assessment of captured drone data from the wreckage and through matching of the acquired data with flight path data acquired from simulated drones in a controlled lab environment.

**Test A:**

1. Make up a flight path using Mission Planner software and save this as a .kml\(^3\) or an NMEA-compatible\(^4\) file format to SatGen Trajectory Generator software\(^5\).

2. Simulate GNSS signal traffic, using an appropriate GNSS almanac, via a LabSat2 GNSS Simulator\(^6\) and a GNSS signal amplifier.

3. Simultaneously record the transmitted GNSS signals on GPSMon Software\(^7\) via

---

\(^2\)The purpose of a Faraday cage is to block external electromagnetic fields, which may interfere with the lab-test signals and skew results. The Faraday cage used in this case is a specially-built secure enclosure within the Security Research Institute at Edith Cowan University, Joondalup, Western Australia.

\(^3\)The Keyhole Markup Language (kml) format is used to create an XML-based file for geographic annotation and visualization purposes on two-dimensional maps and three-dimensional planet browsers (such as Google Earth). developers.google.com (2018) lists the parameters that can be contained within a .kml file.

\(^4\)National Marine Electronics Association (NMEA) file format is used for communication between many types of marine electronic devices, including GPS receivers.

\(^5\)SatGen software (LabSat, 2018b) enables the creation of a GNSS signal-data user-generated trajectory file. This can then be replayed on a LabSat GNSS Simulator.

\(^6\)LabSat2 (LabSat, 2018a) simulates the navigation signals associated with various GPS systems.

\(^7\)GPSMon (Raymond, n.d.) provides real-time GPS packet monitoring and control.
a BU-353S4 USB GNSS Receiver\(^8\) and on the Cheerson CX-20 drone via Mission Planner software.

4. Compare the recorded GPSMon control flight path data with the recorded Cheerson CX-20 drone/Mission Planner flight-path data.

**Test B:**

1. Take a flight data record from the drone (crash flight from 4 Jan 2015 selected), convert it to a .kml or an NMEA-compatible file format, and input it to the SatGen Trajectory Generator software.

2. Generate the GNSS signal traffic, using a GNSS almanac from 4 Jan 2015, for the flight data record selected via a LabSat2 GNSS Simulator and a GNSS signal amplifier.

3. Record the transmitted GNSS signals on GPSMon Software via a BU-353S4 USB GNSS Receiver.

4. Compare the recorded GPSMon control flight path data with the selected Cheerson CX-20 drone flight path data.

### 3.3 Collection of Evidence

#### 3.3.1 Extraction of Residual Flight Data

1. Inspection of the internal components of the CX-20 showed that it consisted of the hardware as published by the drones manufacturer.

2. The drone’s owner had removed the 11.1V 2700 mAh power supply that came with the unit and cut the power cables off at the terminals on the motherboard. Replacement wires were soldered in place, and an MG Power 35C 11.1V 5300 mAh power supply was attached. Restoration of power was successful in that lights within the large white APM box in the center of the drone flashed and one of the rotor motors attempted to turn.

3. The APM box was opened, and the APM v2.52 CPU circuit board on top was separated from the APM I/O circuit board below.

![Figure 3. Top View of the APM v2.52 CPU Circuit Board.](image)

Figure 3. Top View of the APM v2.52 CPU Circuit Board.

![Figure 4. Bottom View of the APM v2.52 CPU Circuit Board.](image)

Figure 4. Bottom View of the APM v2.52 CPU Circuit Board.

4. The APM CPU board was connected to a laptop computer. After installation of the appropriate Arduino Mega 2560 driver, serial communications were

\(^8\)The BU-353S4 (oziexplorer3.com, n.d.) is a small magnet-mounted GPS receiver suitable for lab-test environments.
established, indicated by flashing green receive/transmit lights on the circuit board.

Figure 5. Serial Communication with APM v2.52 CPU Circuit Board Established via Micro USB Cable.

5. To interact with the sensors on the APM CPU board and access any residual flight data in memory, the manufacturer-recommended flight management software Mission Planner was installed on the laptop computer (version 1.3.49 build 1.1.6410.20232).

6. Mission Planner detected the APM CPU board on serial port COM3, to which it successfully connected.

7. The ‘DataFlash Logs’ tab was selected on the Mission Planner GUI, and the ‘Download DataFlash Log Via Mavlin’ button was clicked. The Mission Planner Log Downloader module extracted all available files on the APM circuit board onto the laptop computer.

Figure 6. Mission Planner GUI with APM v2.52 CPU Circuit Board Connected on COM3.

8. A forensic copy of the downloaded log file directory was made, which would be used as a working copy for analysis of the downloaded files. The integrity of the forensic copy was verified by comparing hash values between the copy and the original file.

3.3.2 Simulation of Flight Data

Test A:

Four waypoints were selected on Mission Planner software within the ECU Joondalup campus, which constituted an irregular-quadrilateral flight path to be simulated.

Figure 7. Mission Planner Log Downloader in Progress.

Figure 8. Mission Planner Log Downloader in Progress.
1. A .kml file for the chosen flight path was uploaded to the SatGen Trajectory Generator software.

2. Mission Planner software configuration was modified to disable all pre-flight tests, disable all advanced parameters, and arm the drone for the impending simulated flight.

3. SatGen Trajectory Generator software initially converted the .kml data into an NMEA-compatible file format, and then into a .dsf format for injection into the LabSat2 GNSS Simulator. The generated GNSS signals, using an up-to-date almanac, were amplified via an HD Communications Core HD26001 GNSS Amplifier before being transmitted within the confines of the Faraday cage.

3. GPSMon software via a BU-353S4 USB GNSS Receiver recorded the transmitted GNSS signals from the simulated crash flight.

4. A comparison was made between the recorded GPSMon control flight path data with the recorded Cheerson CX-20 drone flight path data.

Test B:

1. Flight data recovered from the drone (crash flight on 4 Jan 2015) was extracted, converted to a .kml file, and uploaded to the SatGen Trajectory Generator software.

2. SatGen Trajectory Generator software converted the .kml data into an NMEA-compatible file format, and then into a .dsf format for injection into the LabSat2 GNSS Simulator. The generated GNSS signals, using an up-to-date almanac, were amplified via an HD Communications Core HD26001 GNSS Amplifier before being transmitted within the confines of the Faraday cage.

3. A comparison was made between the recorded GPSMon control flight path data with the downloaded crash flight data from the Cheerson CX-20 drone.

3.4 Analysis

The \logs\QUADROTOR\1 directory contained 75 files. Of these several small files had timestamp dates which significantly predated the viable date of manufacture, and these are interpreted to have existed from the time this CX-20 left the assembly plant. More realistic timestamp dates were associated with larger files, which could be used to differentiate between separate flights. Six files with extensions .bin, .kmz, .log, .log.gpx, .log.param, and .log0wp.txt were associated with each of seven separate datasets on 25 Dec 2014, 26 Dec 2014, 28 Dec 2014, 29 Dec 2014, 3 Jan 2015 and two on 4 Jan 2015. Each of these datasets corresponds to an individual drone flight.

The seven .kmz files were imported into Google Earth Pro version 7.1.5.1557. Of particular interest is the second flight on 4 Jan 2015, which shows the CX-20s final sortie and its near-vertical fall to the moment of impact.
In Figure 9, the differently colored vertical lines represent different APM modes for the drone:

1. Orange - ‘Stabilize’ (Manual flight mode
   Responding to pilot commands)
2. Yellow - ‘Loiter’ (Stable/hold position
   no input)
3. Green - ‘RTL’ (Return to Launch mode
   Track to be taken to return to the original take-off position)

Also of interest is the ‘flight’ recorded on 28 Dec 2014. 65 seconds of data exist and are located in the center of the owner’s house (out of respect for the owner’s anonymity, this will not be shown). However, if this had been a criminal investigation, the fact that residual drone data was able to locate a specific address is extremely significant.

### 3.4.1 Crash Investigation deduced from Residual Flight Data

While this paper is primarily concerned with the potential usefulness of forensic analyses of drones to bring criminals to justice, the chance of having a crash-damaged drone to analyze spawned the opportunity of conducting an ‘air-crash investigation’ in an attempt to determine the cause of the drone’s demise. The following two steps were outlined to accomplish this:

1. Analyze all pertinent parameters associated with the drone’s final flight, from take-off to impact.
2. If possible, ascertain the cause of the crash.

Mission Planner software is capable of performing detailed analyses of log files downloaded from the CX-20 post-flight. Such an analysis was performed to determine the cause of the drone crash on 4 Jan 2015.

An online search was made to find historical weather data for that date in the Ocean Reef area. However, most historical data only covered the past 14 months. Being late afternoon local time and during summer, it is reasonable to assume that a moderate to fresh south-westerly sea breeze was blowing at the time the flight was made.

The .kmz file shown above in Figure 8 already shows the moment that the drone crashed. From take-off, it had gained altitude while flying over the water at Ocean Reef Marina (orange lines), until the moment it a technical fault caused it to enter Loiter mode. The yellow line represents the drones fall to impact at the south-western corner of the car park, close to the start of the southernmost boat ramp. The green lines represent the path and altitude that the drone would have taken, without pilot input, to return to the launch position, had it not been for the technical fault.

From the ‘DataFlash Logs’ tab on the main user interface, the ‘Review a Log’ button was clicked. The logs\QUADROTOR\1\2015-01-0409-42-17.bin file was selected. A series of graphs were produced to assess different attributes during the flight.

Since the .kmz file had indicated that the drone had entered ‘loiter’ mode unexpectedly, the ‘copter loiter’ plot was generated on the Mission Planner Log Browser. dVelX, dVelY, VelX and VelY parameters under the...
NTUN (navigation information) folder were displayed.

Figure 10. ‘Copter Loiter’ Parameters Plot.

This plot shows that both lateral and inline motion reduced to zero at 09:43:01 UTC. The assumption is that this is the moment the drone ceased responding to input commands and impacted the car park within a couple of seconds later. The plot also shows that both lateral and inline motion began a trend towards zero at 09:42:56 UTC. Hence the technical fault must have commenced within seconds before this time.

Now that the possible time range when the technical fault and crash could have occurred was known, other parameters were investigated to assess the cause of the crash. Firstly, power failure was able to be ruled out as a cause of the drone crash. Despite battery voltage and battery current not being logged, board voltage was available, which showed that power had been maintained to the main board throughout the flight.

Figure 11. Board Voltage in mV.

Throttle plots provided more information than the battery voltage. ‘Throttle In’ data flat-lined (at ≈37.7% of full power) at approximately 09:42:44 UTC on 4 Jan 2015.

Figure 12. Throttle In (red) vs Angular Boost (green) vs Throttle Out (blue) on a 0-1000 scale.

The three attitude attributes of roll, pitch and yaw were plotted to see whether the issue with the throttle affected the drone’s stability. The plot of pitch angle shows a significant spike coincident with the time the ‘Throttle In’ attribute flat-lines at 09:42:44 UTC on 4 Jan 2015.

Figure 13. Roll (red) vs Desired Roll (green) in centidegrees

Figure 14. Pitch (red) vs Desired Pitch (green) in centidegrees

Figure 15. Yaw (red) vs Desired Yaw (green) in centidegrees

The fact that these graphs show spurious data being logged after the moment of impact in borne out by the altitude plots. GNSS altitude vs barometric altitude show height of 41 m and 49 m respectively at 09:43:01 UTC. However, both altitude sensors do not record the rapid fall of the drone to impact with the car park. The suggestion is that the technical fault had resulted in the loss of parameters being logged accurately at all.

Figure 16. GNSS Altitude (red) vs Barometric Altitude (green) in meters
The physics of accelerated motion tells us that
\[ s = ut + \frac{1}{2}at^2 \quad (1) \]
where:

- \( s \) = distance traveled (m)
- \( u \) = velocity at time zero (ms\(^{-1}\))
- \( a \) = acceleration due to gravity (ms\(^{-2}\))
- \( t \) = time of travel (s)

If we assume the fall occurred in a vacuum from a median height \( s \) of 45 m between the two sensor values we have and assume that at the beginning of the fall there was no vertical motion (i.e. \( u = 0 \) ms\(^{-1}\)), then:

\[ t = \sqrt{\frac{(45 \times 2)/9.81}{2}} = 3.03 \text{ s} \quad (2) \]

To account for some effect from air resistance, let’s say that it took four seconds for the drone to fall from a height of 45 m to its impact on the car park.

To calculate impact velocity, we can use the equation:
\[ v^2 - u^2 = 2as \quad (3) \]
where:

- \( s \) = distance traveled (m)
- \( u \) = velocity at time zero (ms\(^{-1}\))
- \( a \) = acceleration due to gravity (ms\(^{-2}\))
- \( v \) = velocity at impact (ms\(^{-1}\))

For the same assumptions made above, then:

\[ v = \sqrt{(2 \times 9.81 \times 45)} = 29.7 \text{ ms}^{-1} \quad (4) \]

If we again allow for some effect from air resistance, we can estimate that the impact velocity would have been at approximately 25 ms\(^{-1}\).

From this evidence, the timeline of events presented in Table 1 can be drawn up for the air-crash incident.

<table>
<thead>
<tr>
<th>Date</th>
<th>Time (UTC)</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-Jan-15</td>
<td>09:42:17</td>
<td>Start of data logging.</td>
</tr>
<tr>
<td>4-Jan-15</td>
<td>09:42:19</td>
<td>Take-off (from altitude and accelerometer plots).</td>
</tr>
<tr>
<td>4-Jan-15</td>
<td>09:42:44</td>
<td>‘Throttle In’ flat-lines at 37.7% power. Spike in pitch angle. Drone stops gaining altitude. ‘Throttle Out’ becomes more erratic.</td>
</tr>
<tr>
<td>4-Jan-15</td>
<td>09:42:56</td>
<td>Drone commenced losing lateral and forward motion. Pitch angle becomes erratic.</td>
</tr>
<tr>
<td>4-Jan-15</td>
<td>09:43:01</td>
<td>Pitch angle still erratic. Ground speed reduced to zero.</td>
</tr>
<tr>
<td>4-Jan-15</td>
<td>09:43:05</td>
<td>Assumed moment of impact (no sensors logging) at 31.7620° S, 115.7281° E.</td>
</tr>
</tbody>
</table>

The loss of accurate parameter logging just before the crash, and certain flight parameters not being enabled in the pre-flight setup has somewhat restricted what could be determined from this investigation. The eyewitness account by the drone’s owner states that he noticed the drone suffered from a lack of response to control commands a few
seconds before one rotor stopped. The resultant instability caused the drone to flip over and impact top-down onto the car park. Servo logging was not enabled, and so the failure of a rotor during the flight could not be determined from the logged data. However, during pre-flight checks before one of the simulations conducted in the Faraday cage, it was noticed that one of the rotor servos would not spin. Additionally, the spike in pitch angle at 09:42:44 UTC may have been a precursor in attitude instability that eventually led to the drone flipping over a couple of seconds later.

3.4.2 Analysis of GNSS Simulation Results

![Image of Control vs Mission Planner vs Drone Lat/Long Comparison for Crash Flight Simulations]

During Test A, Test B1 and Test B2, the satellite quality control parameters of Horizontal Dilution of Precision (HDOP), Position Dilution of Precision (PDOP), Vertical Dilution of Precision (VDOP), and the number of satellites in view above the elevation mask were within acceptable limits.

3.5 Reporting

The final stage in the OSCAR principle consisted of the completion of this research paper. A running sheet of the forensic process followed is shown in Table 2.

4. CONCLUSIONS

Forensic investigation of UAVs/drones, a key component of smart cities, is a challenging task mainly due to the lack of standardization and weak published research work. Through this research we have demonstrated that the acquisition of drone data, particularly that of a crashed UAV/drone, is possible in a forensically sound manner. The data extracted, predominantly the GNSS data that provided us with launch-point evidence, was subsequently used to validate simulation results obtained through a controlled environment for drone simulation, i.e. undertaken within a Faraday cage. This proved that the method adopted is accurate, sound, and can easily be adopted to other types of UAVs/drones for data verification. This approach may also assist law enforcement agencies to bring sufficient evidence to identify the positional history of the drone during the suspicious act under investigation by forensic investigating officers.

ACKNOWLEDGMENTS

The authors wish to express their sincere gratitude to the owner of the drone, who granted Edith Cowan University, by way of a letter of authority, permission for the drone and any data it contained to be used for research purposes.
Test A (Simulated flight over Edith Cowan University - Control vs Drone):

(a) Control & Drone Lat/Long

(b) Control vs Drone Lat/Long Comparison, and Control/Drone Maps

(c) Control vs Drone Distance Difference & Distance Difference Histogram

Figure 18. Control vs Drone Result for Edith Cowan University Simulation
**Test B1** (Simulated flight from Crash Data - Control vs Mission Planner):

- **(a)** Control vs Mission Planner Lat/Long
- **(b)** Control vs Mission Planner Lat/Long Comparison, and Control/Mission Planner Maps
- **(c)** Control vs Mission Planner Distance Difference & Distance Difference Histogram

Figure 19. Control vs Mission Planner Result for Crash Flight Simulation
Test B2 (Simulated flight from Crash Data - Drone vs Mission Planner):

(a) Control vs Drone Lat/Long

(b) Control vs Drone Lat/Long Comparison, and Control/Drone Maps

(c) Control vs Drone Distance Difference & Distance Difference Histogram

Figure 20. Control vs Drone Result for Crash Flight Simulation
Table 2. Forensic Analysis Running Sheet

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>29-Sep-17</td>
<td>13:00</td>
<td>Inspected internal components of the crash-damaged drone.</td>
</tr>
<tr>
<td>29-Sep-17</td>
<td>13:30</td>
<td>Established power using an 11.1V 5300mAh power supply (MG Power 35C).</td>
</tr>
<tr>
<td>29-Sep-17</td>
<td>14:00</td>
<td>The APM v2.52 circuit board was removed.</td>
</tr>
<tr>
<td>29-Sep-17</td>
<td>14:30</td>
<td>Confirmed serial communications present between the APM board and laptop</td>
</tr>
<tr>
<td></td>
<td></td>
<td>computer via a micro USB cable.</td>
</tr>
<tr>
<td>29-Sep-17</td>
<td>15:30</td>
<td>Installed Mission Planner software, version 1.3.49 build 1.1.6410.20232.</td>
</tr>
<tr>
<td>29-Sep-17</td>
<td>15:45</td>
<td>Connected APM on Mission Planner via serial port COM3.</td>
</tr>
<tr>
<td>29-Sep-17</td>
<td>16:00</td>
<td>Downloading all available log files.</td>
</tr>
<tr>
<td>29-Sep-17</td>
<td>16:10</td>
<td>Forensic copy of downloaded files taken as a working copy. Original files</td>
</tr>
<tr>
<td></td>
<td></td>
<td>stored on a forensically wiped and reformatted Emtec 16Gb USB drive.</td>
</tr>
<tr>
<td>30-Sep-17</td>
<td>09:00</td>
<td>Commenced inspecting .kmz files with Google Earth Pro.</td>
</tr>
<tr>
<td>30-Sep-17</td>
<td>09:30</td>
<td>Seven .kmz files contain flight data. Crash damage determined to have</td>
</tr>
<tr>
<td></td>
<td></td>
<td>occurred at ≈09:32 UTC on 4 Jan 2015. Owner’s residence ‘identified’.</td>
</tr>
<tr>
<td>30-Sep-17</td>
<td>09:45</td>
<td>Commenced investigating of the drone crash.</td>
</tr>
<tr>
<td>30-Sep-17</td>
<td>11:00</td>
<td>Generating graphs of significant flight parameters.</td>
</tr>
<tr>
<td>29-Sep-17</td>
<td>12:30</td>
<td>Crash investigation completed. Commenced write-up.</td>
</tr>
<tr>
<td>1-Oct-17</td>
<td>10:00</td>
<td>Five waypoints selected on ECU Joondalup Campus for generation of</td>
</tr>
<tr>
<td></td>
<td></td>
<td>simulated flight plans.</td>
</tr>
<tr>
<td>1-Oct-17</td>
<td>11:30</td>
<td>Four flight plans generated in text and .kml format.</td>
</tr>
<tr>
<td>1-Oct-17</td>
<td>12:00</td>
<td>One flight plan uploaded to the APM CPU board using Mission Planner</td>
</tr>
<tr>
<td></td>
<td></td>
<td>software.</td>
</tr>
<tr>
<td>2-Oct-17</td>
<td>10:15</td>
<td>Test simulation commenced in Faraday cage at SRI.</td>
</tr>
<tr>
<td>2-Oct-17</td>
<td>11:30</td>
<td>Test simulation completed.</td>
</tr>
<tr>
<td>2-Oct-17</td>
<td>12:00</td>
<td>No GNSS data recorded on CX-20.</td>
</tr>
<tr>
<td>10-Oct-17</td>
<td>13:45</td>
<td>Test simulation commenced in Faraday cage at SRI.</td>
</tr>
<tr>
<td>10-Oct-17</td>
<td>14:15</td>
<td>Mission Planner software installed on the Faraday cage PC to assess</td>
</tr>
<tr>
<td></td>
<td></td>
<td>drone GNSS detection.</td>
</tr>
<tr>
<td>10-Oct-17</td>
<td>15:45</td>
<td>CS-20 determined to be detecting GNSS satellites okay, but not generating</td>
</tr>
<tr>
<td></td>
<td></td>
<td>log files. Test simulation completed.</td>
</tr>
<tr>
<td>16-Oct-17</td>
<td>10:00</td>
<td>Test simulation commenced in Faraday cage at SRI.</td>
</tr>
<tr>
<td>16-Oct-17</td>
<td>10:30</td>
<td>New flight path generated over ECU Joondalup campus and uploaded</td>
</tr>
<tr>
<td></td>
<td></td>
<td>to SatGen Trajectory Generator software.</td>
</tr>
<tr>
<td>16-Oct-17</td>
<td>12:30</td>
<td>Several drone pre-flight tests and advanced parameters disabled to ensure</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the drone arms itself before any simulated flight.</td>
</tr>
<tr>
<td>16-Oct-17</td>
<td>13:00</td>
<td>Mission Planner .rlog and .tlog files saved for the simulated flight</td>
</tr>
<tr>
<td></td>
<td></td>
<td>concurrent with GPSMon files logged.</td>
</tr>
<tr>
<td>16-Oct-17</td>
<td>14:00</td>
<td>Test completed for simulation over ECU Joondalup Campus.</td>
</tr>
<tr>
<td>22-Oct-17</td>
<td>13:40</td>
<td>Graphical presentations of ECU Joondalup Campus simulation data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>generated.</td>
</tr>
<tr>
<td>23-Oct-17</td>
<td>12:20</td>
<td>Crash-flight data from drone converted to .kml file and uploaded to</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SatGen Trajectory Generator software.</td>
</tr>
<tr>
<td>23-Oct-17</td>
<td>12:30</td>
<td>Crash-flight recreated by simulation and recorded by GPSMon control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>software.</td>
</tr>
<tr>
<td>23-Oct-17</td>
<td>12:45</td>
<td>Test completed for simulation using crash-flight data.</td>
</tr>
</tbody>
</table>
REFERENCES


