

Integrating the OpenSky Network into GNSS-R Climate Monitoring Research [†]

Mike Laverick^{1,*}, Delwyn Moller², Christopher Ruf³, Stephen Musko³, Andrew O'Brien⁴, Ryan Linnabary⁴, Wayne Thomas⁵, Chris Seal¹ and Yvette Wharton¹

¹ Centre for eResearch, The University of Auckland, Auckland 1010, New Zealand; c.seal@auckland.ac.nz (C.S.); y.wharton@auckland.ac.nz (Y.W.)

² Department of Electrical, Computer and Software Engineering, The University of Auckland, Auckland 1010, New Zealand; delwyn.moller@auckland.ac.nz

³ Climate and Space Department, Michigan University, Ann Arbor, MI 48109, USA; cruf@umich.edu (C.R.); musko@umich.edu (S.M.)

⁴ Electrosience Laboratory, The Ohio State University, Columbus, OH 43212, USA; obrien.200@osu.edu (A.O.); linnabary.24@buckeyemail.osu.edu (R.L.)

⁵ Air New Zealand, 185 Fanshawe Street, Auckland 1010, New Zealand; wayne.thomas@airnz.co.nz

* Correspondence: mike.laverick@auckland.ac.nz or researchdata@auckland.ac.nz

[†] Presented at the 8th OpenSky Symposium 2020, Online, 12–13 November 2020.

Published: 1 December 2020



Abstract: Global Navigation Satellite System Reflectometry (GNSS-R) provides a unique means of inferring geophysical conditions of the Earth's surface without the need for costly, and often infeasible, in-situ climate monitoring systems. As part of NASA's Cyclone Global Navigation Satellite System (CYGNSS) mission, and in conjunction with Air New Zealand, we are taking the novel approach of mounting a GNSS-R receiver on a commercial aircraft, which shall allow for an unprecedented collection of climate data over and around the islands of New Zealand. Such data include inundation and coastal dynamics, and soil moisture content and variability. We report back to the community how the OpenSky Network data support our climate monitoring research. We discuss how we use the historical database state-vectors to simulate and visualise the predicted geographical coverage of the airborne GNSS-R receiver. We also discuss how the live API can help monitor our payload in-flight, our investigations into the OpenSky ADS-B coverage over New Zealand, and our plans to expand the coverage.

Keywords: state-vector data; data visualisation; OpenSky Network; climate research

1. Introduction

1.1. Global Navigation Satellite System Reflectometry

Global Navigation Satellite System Reflectometry, henceforth GNSS-R, is the process of measuring radio waves emitted by GNSS satellites such as GPS and Galileo, which have reflected off of the surface of the Earth. By measuring the reflected signal one can determine two main aspects of the surface: the dielectric properties and the roughness state of the surface. By measuring the direct signal emitted from the GNSS, and the reflect signal, one can determine these two properties and make inferences about the reflected surface including estimates of soil moisture content over land, and wind speeds over bodies of water [1].

The latter of these abilities led to the development of the NASA Cyclone Global Satellite System (CYGNSS) mission, which uses GNSS-R to pierce through the thick clouds of tropical cyclones and hurricanes and map out their internal structures using the derived wind speeds above the ocean

surfaces [2]. More information on the CYGNSS mission can be found here (<https://www.nasa.gov/cygnss>). CYGNSS helps not only to understand weather fronts, but also to monitor and help predict hurricane landfall location and impact. The CYGNSS mission comprises eight micro-satellites in a low Earth orbit, each with a GNSS-R receiver capable of receiving four simultaneous GNSS signals. As these satellites orbit around the Earth, they are able to combine their reflection measurements to build up a picture of oceanic wind speeds, and thus map out strong weather events. In addition, the GNSS-R measurements can not only be used to map out cyclone structures, but they can also be used to monitor land-based properties and events. Ref. [3] highlights some of the potential uses of the CYGNSS data, including mapping out jungle and rainforest waterways despite thick vegetation [4], monitoring soil moisture content [5], and even monitoring on-going flooding and inundation caused by natural disasters [6].

1.2. NGRx: The Next Generation GNSS-R Receiver

The CYGNSS mission has been so successful in hurricane monitoring, and in exploring other novel applications of the data, that NASA has agreed to extend the mission through 2023 and to develop the next generation of GNSS-R receiver, henceforth referred to as the NGRx. The NGRx is capable of receiving up to 20 simultaneous GNSS signals at once, and can capture data at a much higher cadence and spatial resolution than the receivers mounted on the CYGNSS satellites.

To complement the ongoing CYGNSS mission, whose low Earth orbits only cover the north-most regions of New Zealand, NASA has partnered with Air New Zealand in the novel effort of installing the science NGRx instrument onboard a routinely-flying commercial aircraft, with the goal of taking GNSS-R measurements during routine flight operations. Hosting such scientific equipment on a commercially operating craft sets an exciting precedent, allowing researchers to piggyback on existing flight operations rather than setting up and operating their own dedicated aircraft and flight routes, and at the same time providing airline operators with the opportunity to contribute in a positive way towards climate monitoring and research—with environmental impact being one of the chief concerns of the aviation industry in today's world.

Initially the NGRx will be mounted on a single commercial Q300 craft in the Air New Zealand fleet, and will be flown several times a day in accordance with the Q300's usual schedule. Two GNSS-R antennas will be installed on the top and bottom of the craft, allowing the under-aisle NGRx to receive both direct GNSS signals from satellites and reflected GNSS signals from the Earth's surface. Recorded GNSS-R data, and NGRx engineering data, will be transmitted back to Payload Operation Centres at the University of Auckland and the University of Michigan via a cellular modem when the aircraft has landed. The NGRx itself is currently under manufacture and development, in a collaborative effort between partners at NASA, the University of Michigan, Air New Zealand, and the Civil Aviation Authority of New Zealand, to ensure that the NGRx has as minimal an impact on flight operations as possible while still producing valuable data for researchers.

2. Integrating the OpenSky Network Data into Our Research

2.1. Use Cases for the OpenSky Network Data

Development of the physical payload is currently underway, but in the interim we can also develop our virtual capabilities at the Payload Operation Centres in the form of data simulation, visualisation, and pipeline codes.

In order to simulate an aircraft-mounted GNSS-R payload we need a number of data sources such as digital elevation maps (DEMs); land cover and coastal maps; GNSS satellite orbital data; antenna electric field pattern data; and of course aircraft flight data. The simulation code was developed using a sample month of purchased FlightAware state-vector data for a single Air New Zealand Q300 craft. However, in order to investigate the coverage on a longer timescale of a year, and for many more aircraft, one would either have to spend substantially more on FlightAware state-vectors; project the

existing month's-worth of data onto other months using purchased flight schedules and assuming identical flight paths per route (a method we investigated prior to the final option); or use the OpenSky Network historical database [7].

We set out to investigate the OpenSky Network with a number of goals in mind, each with different levels of priority:

- minimum: to obtain a month's-worth of state vectors for a single craft for comparison purposes;
- preferred: to also obtain API access to retrieve live tracking data for a single craft;
- ideal: to obtain a year's-worth of state vectors for all 22 Q300 craft in Air New Zealand's fleet and API access for live tracking of aircraft.

Our minimum aim was to obtain a month's-worth of state-vector data for a single Q300 craft, for the purposes of comparing the state-vector data quality with that of the FlightAware data that were previously purchased. Our preferred outcome was to also be able to obtain an API hook to facilitate tracking the aircraft on which we will mount the NGRx. While the NGRx will be in contact with the Payload Operation Centre each time the craft lands, we are otherwise ignorant of the payload's location without referring to flight schedules or flight tracking mechanisms. Our ideal goal would be to also obtain a year's-worth of state-vector data for all 22 Q300s in the Air New Zealand fleet, which would allow us to develop realistic predictions of the quantity and quality of GNSS-R coverage we expect to see from mounting NGRx instruments on multiple craft for long periods of time.

Fortunately, the OpenSky Network is able to fulfil all three of the above goals through use of the Historical Database and the Live REST API, and so we shall be able to realise all of our research goals relating to both coverage predictions and ongoing flight monitoring.

2.2. Data Retrieval and Processing

We queried the OpenSky Network historical database for all state-vector data pertaining to the 22 Q300 craft in the Air New Zealand fleet, over the period 1 January 2019 through to 31 December 2019. The query was set up to retrieve all available Q300 table data from the *state_vectors_data4* table, specifying the Q300s using their ICAO24 transponder IDs, split up on a day-by-day retrieval resulting in 365 separate queries to the database. The data retrieved via the Impala shell were piped directly to 365 separate CSV files, as described by the OpenSky historical database guide (<https://opensky-network.org/data/impala>), resulting in 20 GB of state-vector data.

The 365 mixed-ICAO24 daily CSV files were converted into 22 single-ICAO24 year files using Python 3.7 (<https://www.python.org/>), and saved as pickled (<https://docs.python.org/3/library/pickle.html>) Pandas dataframes (<https://pandas.pydata.org/> version 0.24.2 [8]). The original method and format of querying all Q300s per day were designed to minimise the number of queries on the historical database, rather than visiting the hourly-structured database tables multiple times to retrieve individual craft data. We also purged the flight data of any state-vectors containing NULL values for *latitude* or *longitude*, and removed one spurious state-vector with an apparently corrupted timestamp of *time* = 1c574888699.

For each of the 22 ICAO24 files, we attempted to separate the year's-worth of data into individual flights in preparation for our science simulations. Initially we attempted to use the *onground* flag to define the start and end of a given flight, but it soon became apparent that lack of coverage near airports (particularly at ground-level as only a few airports, such as Kerikeri and Gisborne, are completely out of OpenSky coverage) and sporadic mid-flight false-positives made using the *onground* flag difficult. Instead we took the simpler approach of defining flights according to any temporal gaps in data of 30 min or longer, which is consistent with the typical turn-around times we would expect for the craft. Performing the flight-splitting resulted in ~50,000 individual flights across all 22 craft for the year of 2019.

Once we had the OpenSky data in the form of individual flights, we could attempt to derive the start and end airports for each of the ~50,000 flights, with the intention of supplementing the real

flight data with extrapolated and interpolated synthetic data to pad out the flight data in preparation for use by our GNSS-R simulation codes.

2.3. Estimating Flight Paths from the State-Vector Data

We attempted to derive start and end airports for the OpenSky flights using purely the state-vector data rather than supplementing the data with any schedule knowledge from other sources. The top left panel of Figure 1 shows the OpenSky flight data for a flight from Auckland (NZAA) to Kerikeri (NZKK) on 1 January 2019 for the craft ICAO24 = C81BB3.

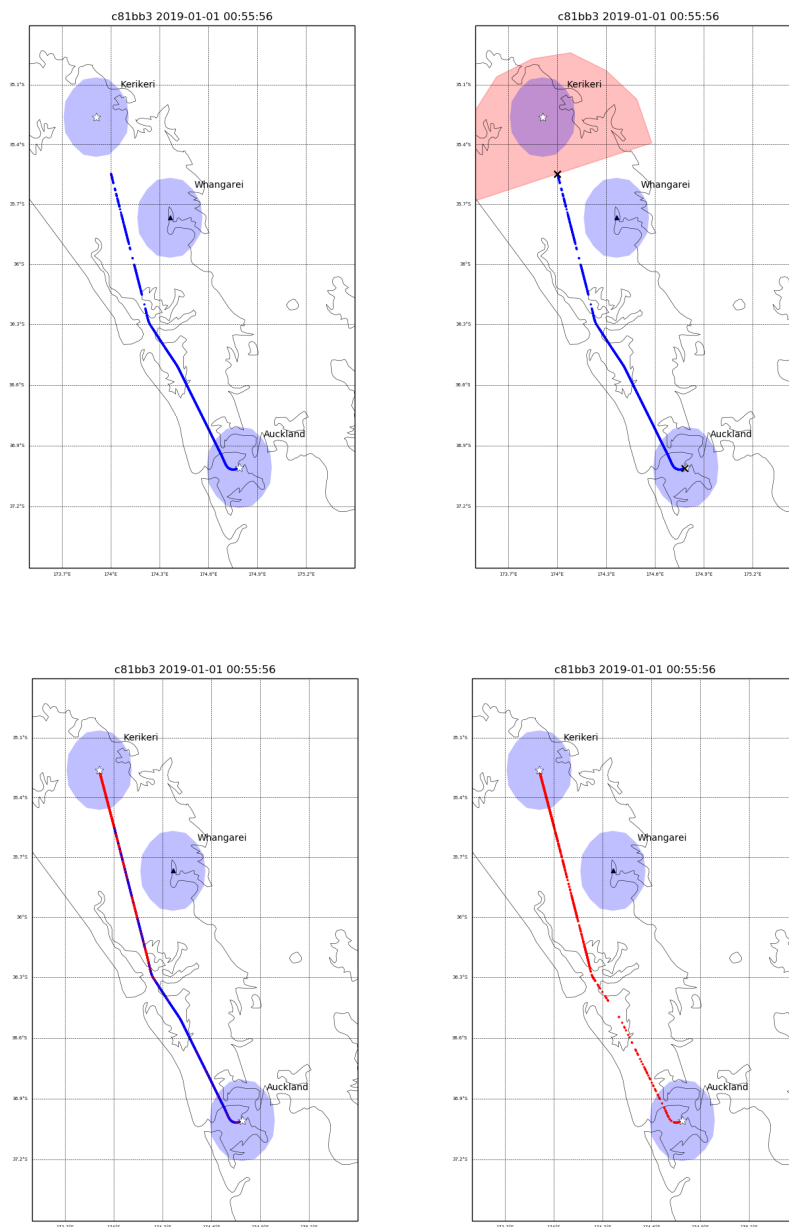


Figure 1. An example of how the state-vector data for an OpenSky flight was extrapolated and interpolated ready for science processing. **(Top left)** A plot of the coordinates (in blue) of a flight from Auckland to Kerikeri. **(Top right)** A schematic of start and end airport derivation for the flight. Blue circles denote a 2 km radius around an airport and the red semicircle denotes the field of view (extending to infinity), based upon craft heading, in which destination airports must fall. **(Bottom left)** The full flight data extrapolated between airports and interpolated down to a 1 s cadence. **(Bottom right)** Only extrapolated and interpolated data for the flight, shown in red.

We used two main methods to determine start and end airports for a flight: (1) if a flight started or ended within ~ 2 km of an airport we considered that to be the start or end airport irrespective of heading. This was to account for any alignments with the runway a craft must make prior to landing. The top right panel of Figure 1 demonstrates this for the start of the journey, which fell within the airport radius of Auckland Airport. (2) If the flight started or ended outside of the ~ 2 km radius around airports, we then used the flight's heading and coordinates to determine the nearest start or end airport. Valid airports for a flight's origin must be in a 180° arc behind the flight starting point, i.e., the opposite direction to the flight's start heading, and valid airports for a flight's destination must be in a 180° arc in front of the flight end point, i.e., the direction of the flight's end heading. Once the potential airports were narrowed down based upon heading we then took the nearest airport in the relevant direction as the start or end airport. The top right panel of Figure 1 depicts the forward-facing arc, which was used to determine the possible end airports. In this case the Whangarei airport was ruled out as a candidate, despite the fact that one can easily imagine a scenario where it would fall closer to a similar flight whose state-vector data ended earlier, thus making Whangarei the closer airport.

Once start and end airports were estimated using the state-vector data, we then extrapolated the existing state-vectors out to the start and end airports. For our purpose of only estimating GNSS-R coverage we just performed a simple linear interpolation in *latitude*, *longitude*, *velocity*, and *vertical_rate*, but one could make more realistic extrapolations if required. In addition to extrapolating the data we also interpolated the data down to a cadence of one second to better represent the GNSS-R data collection rate. The bottom left panel of Figure 1 shows the full flight data post-interpolation with the original OpenSky data shown in blue, and the extrapolated and interpolated data shown in red. The bottom right panel of Figure 1 shows only the extrapolated and interpolated flight data. We performed the airport estimation and flight extrapolation and interpolation for all $\sim 50,000$ flights, and saved the expanded state-vectors once more in the form of pickled Pandas dataframes ready for ingestion into the GNSS-R calculations.

3. Visualising and Assessing the OpenSky Network Data

3.1. Visualising Historical GNSS-R Flight Data

Now that we have a reasonably complete set of historical flight data for many aircraft throughout 2019, we can pass the flight data into our GNSS-R simulation codes and predict the quality of data, quantity of data, and surface coverage we expect to produce when the NGRx is finally installed and flying over New Zealand. Our investigations into expected GNSS-R coverage will allow us to determine environmentally interesting regions (such as wetlands, forests, flood plains, and more) with high predicted coverage, and thus allow us to determine a number of geographical sites that would make good candidates for complementary soil moisture sensor installation. These additional installations will allow us to cross-calibrate the NGRx with the in-situ measurements, and even help cross-calibrate between additional missions such as CYGNSS.

As well as predicting expected yearly coverage for the purposes of choosing cross-calibration sites, the Opensky Data and their corresponding simulation results can be used to develop historical data visualisation capabilities in preparation for the data the NGRx will produce. Each flight will produce a wealth of data containing information on geospatial coverage over a wide range of locations and terrain types, tied together with reflected signal strength and measurement quality, which will be made publicly available through a number of data repositories (such as the NASA Physical Oceanography Distributed Active Archive Center (PO.DAAC: <https://podaac.jpl.nasa.gov/>)). To help explore the data that will be produced by the NGRx, we are developing interactive geospatial visualisation tools to help explore and playback data. By using the Opensky-driven simulations we can begin to develop these tools well in advance of our first flight.

Figure 2 shows a snapshot of the flight visualisation tool currently in development, created using the Opensky flight data and GNSS-R simulation code. A video example of the full flight can

be found here (*flight visualisation: AKL to KKE* <https://www.youtube.com/watch?v=GbmFFjoQtr8>) and here (*flight visualisation: KKE to AKL* <https://www.youtube.com/watch?v=xcSlaXDtd8w>). It is our intention to allow users to explore the data both spatially and temporally, in addition to flight playback features. The visualisation tool was written in Python 3.7, makes use of the Bokeh library (<https://bokeh.org/>), and will eventually use the Datashader package for multi-million data points rastering (<https://datashader.org/>).



Figure 2. Snapshot of the flight visualisation tool currently in development. The visualisation shows an OpenSky flight (in yellow) along with the corresponding simulated specular point reflections (in pink) produced as the flight progresses. Video examples of the full flight can be found here (<https://www.youtube.com/watch?v=GbmFFjoQtr8>) and here (<https://www.youtube.com/watch?v=xcSlaXDtd8w>).

3.2. Visualising OpenSky Network ADS-B Coverage over New Zealand

While estimating the full flight paths of the historical OpenSky data discussed in Section 2.3 it became apparent that the OpenSky Network ADS-B coverage in New Zealand was not perfect. One of our main airports of interest, Kerikeri Airport located in the Northland region of New Zealand, appears to fall just outside of the ADS-B coverage, exacerbated by the fact that the Whangarei airport lies just within the ADS-B coverage along the same flight path used for Kerikeri. A similar out-of-coverage issue was also seen for Gisborne, to the east of the North Island, for which the southerly flight paths unfortunately stop around Napier and thus make airport determinations difficult.

We also noted that there appeared to be temporal issues in the OpenSky network coverage, as shown by Figure 3, throughout 2019 in which coverage in the central region of New Zealand dropped out for a number of months. We performed an additional OpenSky historical database retrieval to investigate the situation in August 2020, which showed that coverage was mostly returned to central New Zealand, albeit it was slightly patchy above the ocean between the two islands. Coverage over the Queenstown area of the South Island remains unknown to this investigation, as the Q300 flights do not operate there.

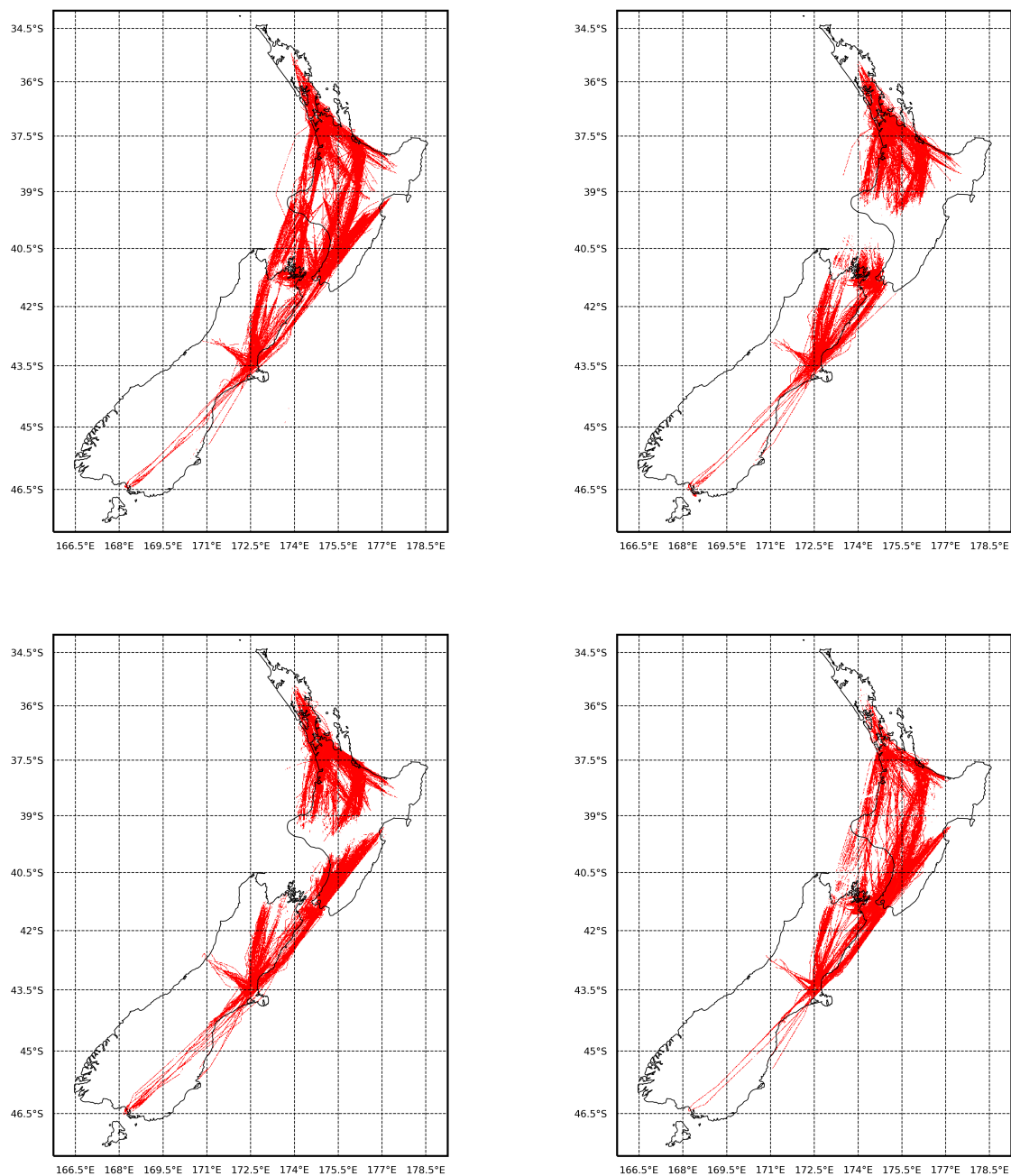


Figure 3. OpenSky Network Q300 data coverage maps over New Zealand during the periods of January 2019 (**top left**), August 2019 (**top right**), December 2019 (**bottom left**), and August 2020 (**bottom right**). We found that the OpenSky network covers the Auckland and Christchurch regions fairly well, but does not quite extend as far eastwards as Gisborne or as far northwards as Kerikeri. Coverage of the South Island seems sufficient to capture the limited quantity of South Island Q300 flights. There also appear to have been temporal coverage gaps over the central North Island during the period of 2019.

4. Discussion and Future Plans

The OpenSky Network looks to be an extremely promising source of flight data, both historical and live, for our climate monitoring needs. Basic flight tracking has been implemented using the OpenSky Live API in conjunction with the basic flight visualisation tools already developed, with the intent to develop this further into a diagnostic dashboard for our payload by integrating it with the received data payload once operational. The live API usage is currently limited to flight tracking,

but we intend to try and connect the API to the GNSS-R simulations for pseudo-live predictions of the data coverage and quality we expect to receive from a tracked flight. These predictions will allow us to prioritise the processing of data from a flight that we believe may have covered areas of interest, such as areas of ongoing flooding and coastal activity. Given the results of our investigation into the OpenSky ADS-B coverage we have decided, admittedly driven by our own interests and need for expanded coverage, to support and expand the network by installing and supporting a number of new receivers around the North Island of New Zealand. Exact locations and numbers are yet to be determined, but we intend to install receivers in the Northland, Gisborne, and Central North Island regions to expand coverage and improve network redundancy. We look forward to our continued use of, and contribution towards, the OpenSky Network.

Author Contributions: Conceptualisation, C.R., D.M., S.M., and A.O.; methodology, C.R., D.M., S.M., A.O., and R.L.; software, S.M., A.O., R.L., M.L., and C.S.; validation, S.M., A.O., and R.L.; investigation, A.O., R.L., M.L., and C.S.; resources, S.M., A.O., R.L., and W.T.; data curation, M.L.; writing—original draft preparation, M.L.; writing—review and editing, M.L.; visualisation, M.L.; project administration, D.M., C.R., and Y.W.; funding acquisition, D.M., W.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Business, Innovation and Employment (grant number UOAX1942).

Acknowledgments: We would like to thank the OpenSky Network team for granting us access to the historical database and for their efforts to provide such an extensive collection of flight data to researchers around the world. We would also like to thank the Bokeh library development team, without whom our ongoing data visualisation efforts would not be possible.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Garrison, J.; Katzberg, S.; Hill, M. Effect of sea roughness on bistatically scattered range coded signals from the global positioning system. *Geophys. Res. Lett.* **1998**, *25*, 2257–2260.
2. Ruf, C.S.; Atlas, R.; Chang, P.S.; Clarizia, M.P.; Garrison, J.L.; Gleason, S.; Katzberg, S.J.; Jelenak, Z.; Johnson, J.T.; Majumdar, S.J.; et al. New ocean winds satellite mission to probe hurricanes and tropical convection. *Am. Meteor. Soc.* **2016**, *97*, 385–395.
3. Ruf, C.S.; Chew, C.; Lang, T.; Morris, M.G.; Nave, K.; Ridley, A.; Balasubramaniam, R. A New Paradigm in Earth Environmental Monitoring with the CYGNSS Small Satellite Constellation. *Sci. Rep.* **2018**, *8*, 8782.
4. Jensen, K.; McDonald, K.; Podest, E.; Rodriguez-Alvarez, N.; Horna, V.; Steiner, N. Assessing L-Band GNSS-Reflectometry and Imaging Radar for Detecting Sub-Canopy Inundation Dynamics in a Tropical Wetlands Complex. *Remote Sens.* **2018**, *10*, 1431.
5. Chew, C.; Small, E. Soil Moisture Sensing Using Spaceborne GNSS Reflections: Comparison of CYGNSS Reflectivity to SMAP Soil Moisture. *Geophys. Res. Lett.* **2018**, *45*, 4049–4057.
6. Chew, C.; Reager, J.T.; Small, E. CYGNSS data map flood inundation during the 2017 Atlantic hurricane season. *Sci. Rep.* **2018**, *8*, 9336.
7. Schäfer, M.; Strohmeier, M.; Lenders, V.; Martinovic, I.; Wilhelm, M. Bringing up OpenSky: A large-scale ADS-B sensor network for research. In Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks, Berlin, Germany, 15–17 April 2014.
8. McKinney, W. Data Structures for Statistical Computing in Python. In Proceedings of the 9th Python in Science Conference, Austin, TX, USA, 28 June–3 July 2010; pp. 56–61.

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).