

# UAV-Based Coverage Measurement Method for 5G

Valentin Platzgummer, Vaclav Raida, Gerfried Krainz, Philipp Svoboda, Martin Lerch, Markus Rupp

*Institute of Telecommunications*

*Technische Universität Wien*

Vienna, Austria

{firstname.lastname}@tuwien.ac.at

**Abstract**—The ability to perform systematic, automatized, and repeatable measurements over large areas is essential for analyzing the spatiotemporal properties of key performance indicators in cellular mobile networks. A systematic spatiotemporal analysis is required to verify 3D channel models and antenna patterns (e.g., massive MIMO or 3D beamforming in 5G). To this end, using unmanned aerial vehicles (UAVs) for measurements opens new possibilities for developers to perform measurements more efficiently, especially now that UAVs have become increasingly available and commercialized.

Currently, researchers employing UAVs to investigate mobile network parameters rely on manual UAV control, which cannot perform repeatable flight routes. In this paper, we introduce our measurement setup using a hexacopter, and then present the initial results of our autonomous measurements.

The stock drones, available on the market, do not fully match the requirements for repeatable measurements; thus we upgraded the drone with additional sensors. We give an overview of the different hardware and software components that we used in our experiment. Accordingly, our results will help other researchers to determine which UAV solutions will be useful in measuring KPIs and to identify the limitations of using this method.

**Index Terms**—unmanned aerial vehicle, drone, hexacopter, autonomous, measurement, wireless, mobile, cellular, cellular mobile network, 5G, LTE

## I. INTRODUCTION

In the last few years, the number of studies that have used unmanned aerial vehicles (UAV) literally exploded. UAVs can be employed in public safety services, precision agriculture, infrastructure inspection, or aerial photography.

From the perspective of cellular mobile networks, the UAV acts either as a relay [1], [2] that provides network coverage (a flying base station, improving connection availability, and increasing capacity) or as a user that demands resources in order to upload collected data or receive new commands.

We focus on the second case, in which the UAV is a user equipment (UE) connected to an operational cellular mobile network. Current networks are being optimized for terrestrial users. The major difference is that the UAV can move more freely in the third dimension, which brings new challenges such as height- and angle-dependent channel models or height-dependent intercell interference.

Researches need to perform repeated UAV measurements to verify the developed models and set configurations (e.g., 3D beamforming in 5G). Numerous field trials from 2G to LTE-A are available (Section I-A) to provide valuable insights.

However, all of the measurement campaigns that we are aware of have been conducted only through manual human

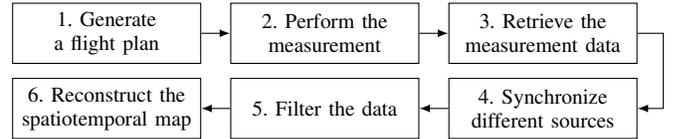


Figure 1. Flowchart illustrating the individual steps of our autonomous UAV measurement setup with automatically generated flight plans.

control of the measuring UAV or through manually designed flight plans. To our knowledge, this is the first publication that describes a repeatable and systematic automatized UAV measurements in wireless networks.

With automatized measurements, we can easily maintain a constant speed and the drone’s orientation at all points of interest, repeat the same flight route regardless of its complexity, and systematically scan large areas (similar to an  $xy$ -table in [3] but in bigger scale) that require battery exchange or recharge. After flight interruption, the UAV continues from the last measured point. Fig. 1 summarizes the whole procedure, which we detail in the rest of the paper.

### A. Related Work

3GPP TR 36.777 [4] has studied the performance of UAVs in LTE Release-14 and has identified several problems (e.g., increased interference compared to terrestrial users) and the corresponding areas for improvement. The document has also introduced various channel models and has presented simulation and field trial results.

In [5], the authors reviewed over 200 research articles related to UAVs in mobile networks. They covered pathloss models, energy consumption models, trajectory models, and empirical results, including field trials or route optimization experiments for UAV access points.

Sundqvist [6] built an LTE-ready drone and subsequently measured the RSRP and RSRQ<sup>1</sup> at an altitude of up to 74 m in a live LTE network. On the other hand, [8] examined the RSSI<sup>2</sup> for an altitude of up to 500 m of GSM and UMTS/HSDPA networks utilizing a human-crewed aircraft, a fixed-wing UAV, and a captive balloon. Additionally, they have introduced an empirical height-dependent channel characterization.

Afonso et al. [2] have discussed the current available communication technologies and their limitations, and have

<sup>1</sup>Reference signal received power, reference signal received quality [7].

<sup>2</sup>Received signal strength indicator.

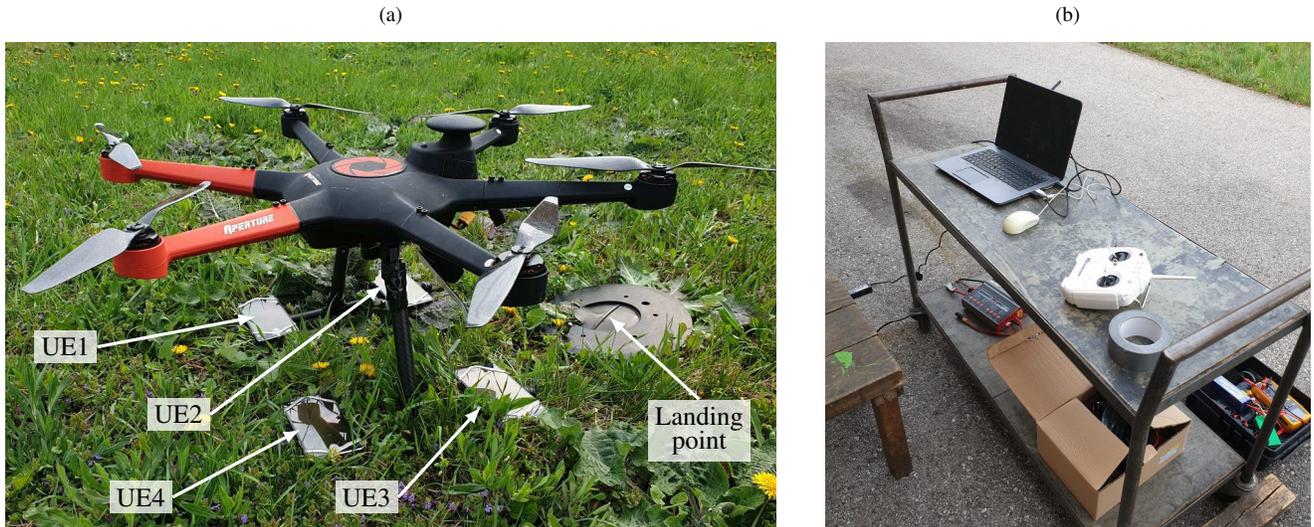


Figure 2. Drone and Ground Control Station: Aperture hexacopter aerial photography drone (a) with four pieces of NEMO user equipment (UE1–4). Ground control station (b): A laptop that is running the QGroundControl software and has a SiK-radio attached.

summarized the protocols for implementing the remote operation of UAVs. In practical experiments, they measured (amongst others) the received power and round trip time in EDGE, HSPA+, and LTE. Meanwhile, [1] investigated the impact of UAV UEs and UAV eNodeBs on the overall LTE performance. Apart from performing simulations, they also conducted interference measurements using a sport airplane and a quadrotor UAV up to the height of 300 m above ground. They concluded that a few of the LTE UAVs can already cause severe interference to ground eNodeBs and ground UEs.

Lin et al. [9] conducted simulations and LTE UAV flights at an altitude of 50 m and 150 m to record various key performance indicators (KPIs), including RSRP and RSRQ. They observed elevated interference levels and handover errors caused by the increased mobility of UAVs. The 3GPP report [10] has investigated the handover failures and the number of detected cells in LTE at heights between 0 and 100 m using UAV UEs. Their results show that the number of handover failures increases with altitude.

In [11], the authors measured LTE interference in a rural area at night (between 2 and 5 a.m.) using a ground UE and a UAV UE at 100 m above the ground. The results show that the UAV UE causes a significantly higher interference than the UE at the ground level. To reduce interference, the authors have proposed using a cruise height control where the UAV and network negotiate an optimal height.

To suppress interference, [12] investigated the following strategies to reduce interference: interference canceling (IC), antenna beam selection, and open loop up-link power control. In [13], the authors calculated the upper bound of the gains that can be obtained by using strategies as interference reduction combining (IRC). They performed drive tests and drone measurements in a live LTE network and evaluated the gains based on the real-world data.

## B. Paper Outline

In section II, we describe the hardware we used in our measurement campaign and the corresponding functions of each component. We also detail the alternative hardware that can be used and the limitations of our setup. We also detail the software of our drone and ground control station.

Section III presents our algorithm to generate the automated flight route and the algorithm to merge the drone’s measurements (height, speed, etc.) with the network measurements conducted using a UE carried by the drone.

Section IV summarizes the measured received power in LTE 800. We used the automatic procedure to systematically scan a rectangular grid in order to capture the large-scale fading in an area of ca. 15 m × 18 m.

## II. THE DRONE AND THE GROUND CONTROL STATION

Lightweight drones have a low load capacity, and under windy conditions, they cannot hold a steady position or fly stably. We thus chose an Aperture hexacopter aerial photography drone [20] (referenced to as “the drone”) with a weight of 4.8 kg and a maximum payload of 1.5 kg (sufficient to carry several cell phones and additional sensors). The closed water-resistant frame protects the drone’s electronics.

Fig. 2 (a) depicts the drone whereas Fig. 2 (b) shows our ground control station (GCS). The rest of this section discusses the drone’s and GCS’s hardware and software.

### A. Flight Controller and Telemetry

The drone comes with an HKPilot32 [21] flight controller unit (FCU) that is based on the Pixhawk [22] open hardware design and runs on a Unix-based operating system. There are several open-source flight stacks available: ArduPilot [23], PX4 [24], or Paparazzi [25]. We decided to use the ArduPilot open source autopilot software due to the good software documentation and large user community.



Figure 3. Drone’s additional hardware: (a) SiK radios for telemetry [14]. (b) LIDAR-Lite 3 laser rangefinder [15]. (c) PX4FLOW optical flow sensor for non-GPS navigation [16]. (d) Turnigy 9X transmitter [17]. (e) FS-A8S 2.4GHz 8-channel receiver [18]. (f) Battery monitor [19].

The drone is delivered with a remote controller that allows for manual control. However, to achieve a repeatable and automatized measurement procedure commanded by a GCS, we need to employ a telemetry radio. There are several possibilities: SiK [14], WiFi, or Bluetooth radios.

The current setup uses a 500 mW, 433 MHz (suitable for Europe) SiK radio, Fig. 3 (a), for its long-range capacity. However, the disadvantage of using such equipment is it has small baud rate (max. 250 kbit/s) and limited flexibility (one receiver/transmitter pair for each vehicle). In the future, we plan to use WiFi radios, which can control multiple vehicles.

### B. Altitude

By default, ArduPilot processes GPS and air pressure sensor data (using Kalman filter) to estimate the altitude. However, the data from both sensors are very inaccurate. To improve the height estimation, we needed additional sensors.

Initially, we utilized MaxBotix [26] ultrasonic sensors (range of 10 m). Unfortunately, they were not working properly as it was giving poor reflection while the drone was hovering above a grassy area. In the second step, we experimented with an optical sensor TF Mini LiDAR [27]; however, the sensor was causing the drone’s height to fluctuate—probably due to the high latency caused by driver problems.

Our current setup uses a rangefinder LIDAR-Lite 3 [15], Fig. 3 (b), that enables the drone to hold the height accurately and to follow the terrain’s altitude. It would be possible to combine both optical and ultrasonic sensors at the same time. Ultrasonic sensors do not work properly on soft surfaces, whereas optical sensors require diffusive reflection to operate correctly. Thus, specular or transparent faces must be avoided. Combining both types of sensors would lead to a broader range of environments, where the drone can operate.

### C. Navigation

Aside from the GPS module, the drone also utilizes accelerometers. ArduPilot fuses GPS data with accelerometer

data (Kalman filter) to estimate the drone’s position. Solutions that are based on relative position estimates, e.g., ultrasonic beacon system Marvelmind [28] or camera-based optical system OptiTrack [29], are more accurate than (low precision) GPS. However, both of these systems are stationary and must be calibrated at every new location.

We have tested the PX4FLOW [16], Fig. 3 (c), to determine whether it can be an alternative to stationary systems. Its optical flow sensor consists of a low-resolution automotive CCD-chip (which scans the surface below the drone) and an ultrasonic sensor (which measures the distance to the ground). Integrating the sensor’s output delivers a relative position. Since the sensor relays on optical flow, it requires a good contrast and cannot operate on poorly textured surfaces, specular surfaces, or under adverse lighting conditions. The performance can be further enhanced by replacing the PX4FLOW stock ultrasonic sensor with a laser rangefinder.

### D. Manual Control

Many countries prohibit using fully autonomous vehicles; thus, our automated measurement setup still provides the user with the option to use manual control.

In the case of a lost connection, the receiver module should ideally signal this status to the FCU; the FCU then undertakes the appropriate countermeasures such as landing or returning the drone to the home position. The default receiver of our drone does not support this feature; thus, we use an alternative transmitter, Fig. 3 (d), and receiver, Fig. 3 (e).

### E. Power Source

Most of the commercial drones are powered by rechargeable lithium polymer (LiPo) batteries that consist of multiple cells (usually two to eight connected in series) and have a capacity between some hundred to several ten thousand mAh (12 000 mAh in the case of our drone). The voltage of each cell should be within the range of 3.2–4.2 V and should never fall below 3 V. Otherwise, the battery can be permanently

damaged. To avoid undervoltage, a battery monitor [Fig. 3 (f)] connected to the battery’s balancer terminal can be used to monitor the voltage. Once the voltage of any cell falls below a certain threshold, the device starts to beep and flash.

### F. Ground Control Station (GCS)

A manual remote control transmitter sends only basic commands to the drone (e.g., throttle, pitch, roll). In contrast, the GCS (in our case, a laptop with a GCS software) can request more complex tasks such as “go to given GPS position”, “land”, or “take a photo”. The GCS can also be used to calibrate sensors, download log files, or update the flight stack.

ArduPilot documentation contains a list of compatible GCS software. We chose QGroundControl (QGC), which is open source and runs on Windows, Linux, and Android. The functionality can be extended with QML Widgets or Qt C++.

## III. MEASUREMENT PROCEDURE

To measure the KPIs of the cellular mobile network, we employ Keysight’s NEMO [30] cell phones. From the “master” tablet, a user can start remotely an automatized measurement script on a “slave” UE via a Bluetooth link. Only one UE, out of the four UEs mounted on the drone, was active in this campaign. We locked the UE to a single eNodeB to prevent unexpected handovers.

### A. Flight Plan

Repeating the same flight route (i.e., maintaining the same speed and route at every point) multiple times is not possible with manual control. The flight route can be replicated by loading a flight plan file into QGroundControl and then sending it to the drone through a telemetry link.

A user can design a flight plan with QGC graphical interface. However, setting up flight plans manually can be tedious; thus, we implemented a MATLAB script that automatically generates a QGroundControl `.plan` file based on several input parameters. The user specifies the drone’s takeoff point; the coordinates of rectangle vertices (or cuboid vertices); number of grid points per dimension ( $N_x$ ,  $N_y$  in case of a rectangle and  $N_x$ ,  $N_y$ ,  $N_z$  in case of a cuboid); and period  $T$  corresponding to a single grid point.

The script then generates a flight plan, the drone takes off based on this flight plan, visits all  $N$  grid points ( $N = N_x \cdot N_y$  for rectangle and  $N = N_x \cdot N_y \cdot N_z$  for cuboid), returns to the starting location, and finally lands. At each grid point, the drone remains steady for period  $T$ . In addition to duration  $N \cdot T$ , which is the amount of time the drone spends hovering at the defined way points, the user also has to consider the transitions between the consecutive grid points. The main challenge here is estimating the time it takes to deplete the battery (depends among others on the wind conditions). Thus, users always have to be more conservative and have to account for a safety margin.

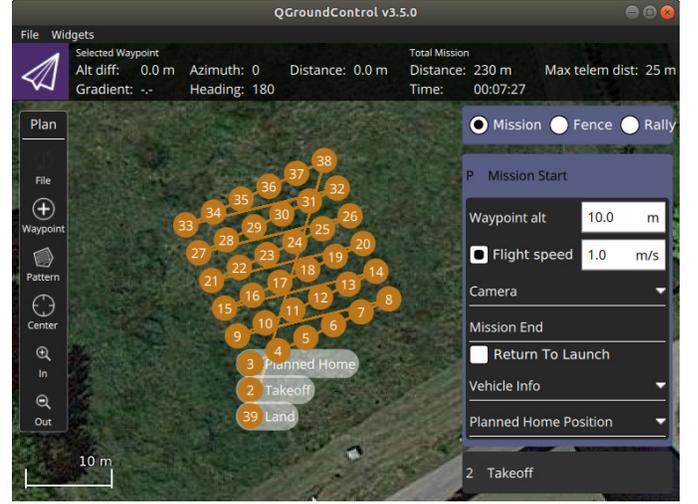


Figure 4. A screenshot capturing a flight plan loaded in the QGroundControl. The flight plan was automatically generated with our MATLAB script.

### B. Measurements Synchronization

The drone logs GPS coordinates, the relative coordinates  $x, y$  (e.g., fused GPS, accelerometer, and flow sensor data using Kalman filter); height  $h$ ; heading  $\phi$ ; and speed  $v$  every  $T_1 \approx 100$  ms. NEMO’s sampling period is approximately  $T_2 \approx 500$  ms. Although there is no time synchronization between the drone and the NEMO UE, we neglect a possible timing offset due to the relatively long 100 ms sampling period.

In what follows, we denote a time stamp by  $t$  and an arbitrary quantity (RSRP, height, speed, etc.) related to this time stamp by  $s$ . To bring the NEMO measurements  $(t_{ue,i}, s_{ue,i})$  and drone’s measurements  $(t_{uav,j}, s_{uav,j})$  to the same time grid, we first shift each NEMO’s time stamps  $t_{ue,i}$  to the nearest drone’s time stamp  $t_{uav,j}$ :

$$\{(t_{ue,i}, s_{ue,i})\}_{i=1,\dots,I} \rightarrow \{(t'_{ue,i}, s_{ue,i})\}_{i=1,\dots,I} : \\ t'_{ue,i} = \arg \min_{t_{uav,j} \in \{t_{uav,j}\}_{j=1,\dots,J}} |t_{ue,i} - t_{uav,j}| \quad \forall i. \quad (1)$$

Then, we replace the drone’s samples  $s_{uav,j}$  at time locations  $t'_{ue,i}$  by the average  $s'_{uav,i}$  of previous  $K = T_2/T_1$  values (in our case,  $K = 5$ ). The number of averaged samples  $s'_{uav,i}$  is thus the same as the number of NEMO’s samples  $s_{ue,i}$ :

$$\{(t_{uav,j}, s_{uav,j})\}_{j=1,\dots,J} \rightarrow \{(t'_{ue,i}, s'_{uav,i})\}_{i=1,\dots,I} : \\ \forall i \exists j: t'_{ue,i} = t_{uav,j}, \quad s'_{uav,i} = \frac{1}{K} \sum_{k=0}^{K-1} s_{uav,j-k}. \quad (2)$$

The  $i$ th NEMO sample,  $s_{ue,i}$ , represents the average of a given quantity (RSRP, throughput, SINR, etc.) within the time interval  $(t_{ue,i-1}, t_{ue,i}]$  with  $t_{ue,i} - t_{ue,i-1} \approx T_2$ . Therefore, we average  $K$  samples  $s_{uav,j}, \dots, s_{uav,j-K+1}$  that are related to the period  $t_{uav,j} - t_{uav,j-K+1} \approx KT_1 = T_2$  with  $t_{uav,j} = t'_{ue,i}$ . We can obtain a worst-case mismatch by shifting  $t_{ue,i}$  to the nearest  $t_{uav,j}$  is  $|t_{ue,i} - t'_{ue,i}| = 50$  ms.

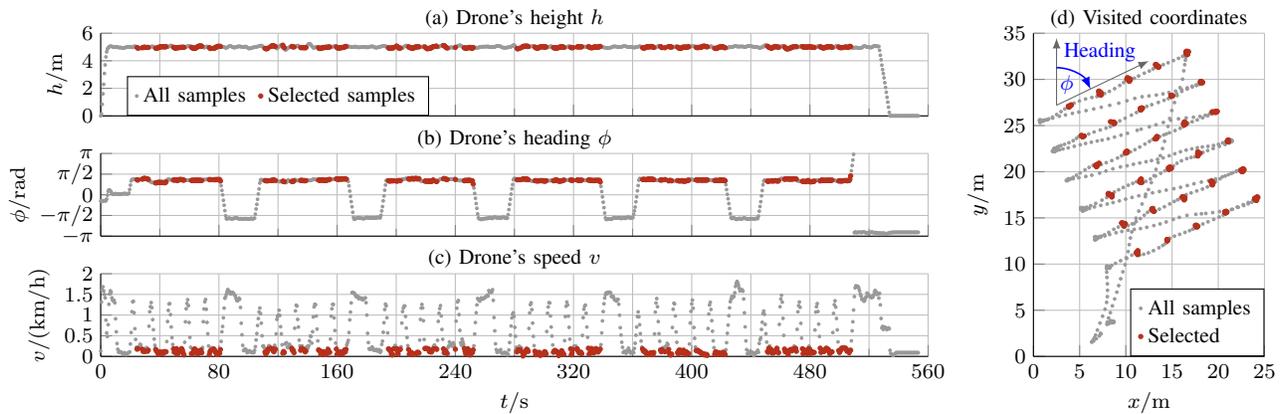


Figure 5. Measurement filtering: Plots (a)–(c) visualize the time series of the drone’s height, heading, and speed during the measurement campaign. Scatter plot (d) shows all visited coordinates. For the RSRP map reconstruction, we consider only those time intervals that fulfill the filter criteria (3) marked in red.

#### IV. MEASUREMENT RESULTS

In this section, we present the results of one of the first measurement flights we conducted. The goal is to measure the RSRP of a single eNodeB on a rectangular grid such that we can perform a spatial analysis of the fading effect. The campaign took place on February 15, 2019 near Klingenbach, Austria in LTE band 20 (LTE 800, 20 MHz bandwidth). The whole flight took approximately 9 minutes.

Fig. 4 shows the flight plan that contains a  $6 \times 6$  measurement grid (way points 3, 4,  $\dots$ , 38).<sup>3</sup> At each way point, the drone is supposed to stop and hover for  $T \approx 7$  s. After finishing a row, the drone flies diagonally to the beginning of the next row, such that it scans each row with the same orientation (heading). Allowing different headings at different measurement points would lead to inconsistent results due to UE’s direction-dependent antenna characteristic.

##### A. Measurement Filtering

Fig. 5 (a)–(c) visualize the time series of measured height above ground  $h_i$ , heading<sup>4</sup>  $\phi_i$ , and speed  $v_i$ . Fig. 5 (d) depicts the visited coordinates  $\mathbf{x}_i = (x_i, y_i)^T$ .

We are interested only in static measurements at the defined grid points, at defined height  $h \approx 5$  m, and with the heading in the main flight direction  $\phi \approx 65^\circ$ ; thus, we exclude all samples (indexes  $i$ ) that do not fulfill the following criteria:

$$45^\circ < \phi_i < 85^\circ \quad \wedge \quad v_i < 0.2 \text{ ms}^{-1} \quad \wedge \quad h_i > 4.5 \text{ m.} \quad (3)$$

The valid samples are marked in red in Fig. 5 (a)–(d).

Here we exclude the first column (way points 3, 9, 15,  $\dots$ , 33 in Fig. 4) due to the wrong heading of the drone after arriving from the previous row. For the additional flights, we need to add a “zeroth” column. In the waypoints of the zeroth column, the drone does not have to wait and hover; it just passes the grid point in order to end up in the points of the first column with the correct heading.

<sup>3</sup>The remaining numbers correspond to taking off, returning, and landing.

<sup>4</sup>Heading is the angle between the north direction and the drone’s front (measured clockwise). The angle  $\phi$  is illustrated in blue in Fig. 5 (d).

##### B. Clustering and Coverage Map Interpolation

In Fig. 5 (d), we obtain a  $6 \times 5$  grid (red points) with approximately 13 RSRP samples per way point. Each way point consists of a cluster of samples with varying positions. The first reason for this is that the drone is never perfectly steady due to wind gusts. The second reason is that the estimated location is corrupted by measurement noise.

In general, there are  $L$  clusters  $\mathcal{C}_l = \{\mathbf{x}_{l,1}, \dots, \mathbf{x}_{l,M_l}\}$  of size  $M_l$ ;  $l \in \{1, \dots, L\}$ . We denote  $M = \sum_{l=1}^L M_l$  as the total number of samples in all clusters. For the  $l$ -th cluster  $\mathcal{C}_l$ , we calculate the centroid  $\mathbf{c}_l = \sum_{\mathbf{x} \in \mathcal{C}_l} \mathbf{x} / M_l$ . The mean distance  $\bar{r} = \sum_{l=1}^L \sum_{m=1}^{M_l} \|\mathbf{x}_{l,m} - \mathbf{c}_l\| / M$  between the samples and the centroids is in our case ca. 11 cm.

We then evaluate the mean RSRP of each cluster. After applying a natural neighbor interpolation [31], [32] to the cluster centroids and their mean RSRP values, we obtain the RSRP map in Fig. 6 (we rotate the  $x, y$  coordinates by  $25^\circ$  clockwise and shift the lower left corner to the origin, resulting in the transformed coordinates  $x', y'$ ).

#### V. CONCLUSION

We have thoroughly described our setup for automatized UAV measurements in wireless networks. As a proof of concept, we have presented our first results collected from LTE 800. This setup can be applied to characterize large-scale fading patterns such that researchers can verify the accuracy of the current models. Likewise, our setup opens possibilities for developing new and improved models.

Compared to other solutions such as the  $xy$ -table proposed in [33], the drone can move freely, thereby enabling researchers to measure much larger areas, in different environments, at arbitrary locations. The automatic flight planning allows us to generate multiple tiles while we exchange the drone’s accumulator. As such, that the total area is not limited by a discharge time of a single battery.

We have achieved a mean distance  $\bar{r} \approx 11$  cm from the cluster centroids. Due to wind gusts, it is impossible to keep the drone absolutely static while being in the air of an outside

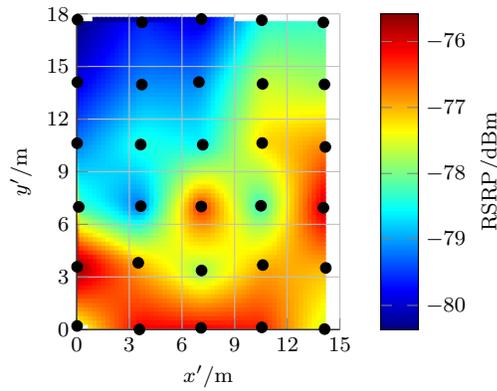


Figure 6. RSRP map regressed with natural neighbor interpolation. Black dots mark the centroids of clusters obtained from filtered coordinates [red points in Fig. 5 (d)].

environment. Additionally, the drone needs to tilt when it attempts to compensate for the wind gust. Consequently, the tilt can impact the received power due to the anisotropic antenna pattern of the UE.

The drone’s random movement at each waypoint does not have to be inherently bad; in an ideal case, it averages out small-scale fading. Caban et al. [33] recommend a  $3\lambda \times 3\lambda$  area. In LTE 800, the wavelength is approximately  $\lambda \approx 37.5$  cm, and thus our diameter  $2\bar{r} \approx 22$  cm does not fulfill this requirement. It is possible to solve this issue by artificially increasing the drone’s hovering area at each waypoint. Hovering slowly within a diameter of  $3\lambda$ , and rotating and tilting to different angles could also help to average out the impact of anisotropic antenna pattern.

So far, we have treated the whole problem as 2D. Since our flight plan generator also supports 3D flight plan routes, our next goal is to analyze the height dependency of the shadowing patterns and to examine 3D antenna patterns.

#### ACKNOWLEDGMENT

This work has been funded by the ITC, TU Wien. The financial support of the Austrian BMFW and the National Foundation for Research, Technology, and Development is gratefully acknowledged. The research has been co-financed by the Czech GA CR (Project No. 17-18675S and No. 13-38735S) and by the Czech Ministry of Education in the frame of the National Sustainability Program under grant LO1401, and supported by the Austrian FFG, Bridge Project No. 871261. We thank A1 Telekom Austria AG for their support and Kei Cuevas for the proofreading.

#### REFERENCES

- [1] B. V. Der Bergh, A. Chiumento, and S. Pollin, “LTE in the sky: trading off propagation benefits with interference costs for aerial nodes,” *IEEE Communications Magazine*, vol. 54, no. 5, pp. 44–50, May 2016.
- [2] L. Afonso, N. Souto, P. Sebastiao, M. Ribeiro, T. Tavares, and R. Marinheiro, “Cellular for the skies: Exploiting mobile network infrastructure for low altitude air-to-ground communications,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 31, no. 8, pp. 4–11, Aug. 2016.
- [3] M. Rindler, S. Caban, M. Lerch, P. Svoboda, and M. Rupp, “Swift indoor benchmarking methodology for mobile broadband networks,” in *86th VTC Fall*, Sep. 2017, pp. 1–5.
- [4] “Study on enhanced LTE support for aerial vehicles,” 3GPP, TR 36.777 v15.0.0, 2017.

- [5] Y. Zeng, Q. Wu, and R. Zhang, “Accessing from the sky: A tutorial on UAV communications for 5G and beyond,” *arXiv e-prints*, p. arXiv:1903.05289, Mar. 2019.
- [6] L. Sundqvist, “Cellular controlled drone experiment: Evaluation of network requirements,” Master’s thesis, Aalto University – School of Electrical Engineering, 2015.
- [7] “Physical layer; measurements,” 3GPP, TS 36.214 v15.3.0, 2018.
- [8] N. Goddemeier, K. Daniel, and C. Wietfeld, “Coverage evaluation of wireless networks for unmanned aerial systems,” in *2010 IEEE Globecom Workshops*, Dec. 2010, pp. 1760–1765.
- [9] X. Lin et al., “Mobile networks connected drones: Field trials, simulations, and design insights,” *CoRR*, vol. abs/1801.10508, 2018. [Online]. Available: <http://arxiv.org/abs/1801.10508>
- [10] KDDI Corporation, “Field measurement results for drone LTE enhancement,” TSG-RAN WG1 Meeting 88bis, 3GPP, R1-1705823, 2017.
- [11] R. Amorim, H. Nguyen, J. Wigard, I. Z. Kovács, T. B. Sørensen, D. Z. Biro, M. Sørensen, and P. Mogensen, “Measured uplink interference caused by aerial vehicles in LTE cellular networks,” *IEEE Wireless Communications Letters*, vol. 7, no. 6, pp. 958–961, Dec. 2018.
- [12] H. C. Nguyen, R. Amorim, J. Wigard, I. Z. Kovács, T. B. Sørensen, and P. E. Mogensen, “How to ensure reliable connectivity for aerial vehicles over cellular networks,” *IEEE Access*, vol. 6, pp. 12 304–12 317, Feb. 2018.
- [13] I. Kovacs, R. Amorim, H. C. Nguyen, J. Wigard, and P. Mogensen, “Interference analysis for uav connectivity over LTE using aerial radio measurements,” in *86th VTC Fall*, Sep. 2017, pp. 1–6.
- [14] PX4 Dev Team. (2019) SiK radio – PX4 v1.9.0 user guide. [Online]. Available: [https://docs.px4.io/en/telemetry/sik\\_radio.html](https://docs.px4.io/en/telemetry/sik_radio.html)
- [15] RobotShop. (2019) LIDAR-Lite 3 laser rangefinder high performance. [Online]. Available: <https://www.robotshop.com/en/lidar-lite-3-laser-rangefinder-high-performance-llv3hp.html>
- [16] D. Honegger, L. Meier, P. Tanskanen, and M. Pollefeys, “An open source and open hardware embedded metric optical flow CMOS camera for indoor and outdoor applications,” in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 1736–1741.
- [17] HobbyKing. (2019) Turnigy 9X 9Ch transmitter. [Online]. Available: [https://hobbyking.com/en\\_us/turnigy-9x-9ch-transmitter-w-module-ia8-receiver-mode-2-afdh-2a-system.html](https://hobbyking.com/en_us/turnigy-9x-9ch-transmitter-w-module-ia8-receiver-mode-2-afdh-2a-system.html)
- [18] —. (2019) FS-A8S 2.4Ghz 8CH mini receiver. [Online]. Available: [https://hobbyking.com/en\\_us/fs-a8s-2-4g-8ch-mini-receiver-with-ppm-i-bus-sbus-output.html?\\_\\_store=en\\_us](https://hobbyking.com/en_us/fs-a8s-2-4g-8ch-mini-receiver-with-ppm-i-bus-sbus-output.html?__store=en_us)
- [19] —. (2019) Lipo voltage checker (2S 8S). [Online]. Available: [https://hobbyking.com/en\\_us/hobbykingtm-lipo-voltage-checker-2s-8s.html](https://hobbyking.com/en_us/hobbykingtm-lipo-voltage-checker-2s-8s.html)
- [20] —. (2019) Aperture Hexacopter Aerial Photography Drone. [Online]. Available: [https://hobbyking.com/en\\_us/aperture-rtf-m1.html](https://hobbyking.com/en_us/aperture-rtf-m1.html)
- [21] PX4 Dev Team. (2019) HKPilot32 – PX4 v1.9.0 user guide. [Online]. Available: [https://docs.px4.io/en/flight\\_controller/HKPilot32.html](https://docs.px4.io/en/flight_controller/HKPilot32.html)
- [22] Pixhawk. (2019) Home page. [Online]. Available: <https://pixhawk.org/>
- [23] ArduPilot. (2019) About. [Online]. Available: <http://ardupilot.org/about>
- [24] Dronecode Project, Inc. (2019) Open source for drones – PX4 open source autopilot. [Online]. Available: <https://px4.io/>
- [25] PaparazziUAV. (2019) Paparazzi UAV. [Online]. Available: [http://wiki.paparazziuav.org/wiki/Main\\_Page](http://wiki.paparazziuav.org/wiki/Main_Page)
- [26] MaxBotix Inc. (2019) Ultrasonic sensors and high performance proximity sensors. [Online]. Available: <https://www.maxbotix.com/>
- [27] DFRobot. (2019) TF Mini LiDAR ToF laser range sensor SKU SEN0259. [Online]. Available: [https://wiki.dfrobot.com/TF\\_Mini\\_LiDAR\\_ToF\\_Laser\\_Range\\_Sensor\\_SKU\\_SEN0259](https://wiki.dfrobot.com/TF_Mini_LiDAR_ToF_Laser_Range_Sensor_SKU_SEN0259)
- [28] Marvelmind Robotics. (2019) Marvelmind Robotics. [Online]. Available: <https://marvelmind.com/>
- [29] NaturalPoint, Inc. (2019) OptiTrack – motion capture systems. [Online]. Available: <https://optitrack.com/>
- [30] Keysight Technologies. (2019) Nemo handy handheld measurement solution. [Online]. Available: <https://www.keysight.com/en/pd-2767485-pn-NTH00000A/nemo-handy>
- [31] R. Sibson, *A brief description of natural neighbor interpolation (Chapter 2)*, V. Barnett, Ed. Chichester: John Wiley, 1981.
- [32] MathWorks. (2019) Interpolate 2-D or 3-D scattered data – MATLAB griddata. [Online]. Available: <https://de.mathworks.com/help/matlab/ref/griddata.html>
- [33] S. Caban, J. A. Garcia Naya, and M. Rupp, “Measuring the physical layer performance of wireless communication systems: Part 33 in a series of tutorials on instrumentation and measurement,” *IEEE Instrumentation Measurement Magazine*, vol. 14, no. 5, pp. 8–17, Oct. 2011.