Application of Path-Based Multipath Component Tracking on Air-Ground Channel Measurement Data

Daniel M. Mielke, Michael Walter, *Senior Member, IEEE*, Dennis Becker, Miguel A. Bellido-Manganell, and Uwe-Carsten Fiebig, *Senior Member, IEEE*

***Abstract*—The operation of unmanned aircraft is unthinkable without reliable wireless communication links: Despite a comparatively high expected level of autonomy of unmanned aircraft, monitoring and remote controlling are required to some degree. As the design of every communication link requires good knowledge on the characteristics of the communication channel, we have performed a measurement campaign to collect channel sounding data for the wireless air-ground channel at C-band. While we have focused on the campaign description and the analysis of the dominant signal component in previous publications, we now concentrate on the detection and tracking of multipath components. In this paper, we present our data processing chain that allows a fast parallel processing of the measurement data, as data dependencies are reduced as much as possible. We furthermore introduce a path-based multipath component tracking approach and apply it to our measurement data. This tracking allows us to estimate the location of reflectors causing multipath component signals. We apply our processing chain to data recorded during take-off at a small airport and compare the results of the reflector localization to a satellite image of the airport to successfully verify our approach.**

***Index Terms*—Channel measurements, channel modeling, channel sounding, multipath components, unmanned aviation, wireless communications.**

I. Introduction

DURING the next years, more and more Unmanned Aircrafts (UAs) are expected to enter the skies. According to [1], the market for UA is expected to have a compound annual growth rate of 12.23 % until the year of 2027. The anticipated fields of application are transportation or other logistic tasks, as well as surveillance, reconnaissance, exploration, and tasks in the agronomy. While UAs are already routinely used in military environments, civil applications, like large unmanned freight aircraft, only emerge. The main task is the integration of civil UAs into non-segregated airspace [2].

One key aspect of this integration is the implementation of a reliable communication link that allows the exchange of Command and Control (C2) data and telemetry data, respectively, between the air vehicle and the remote pilot among other potential communication partners. The type and amount of data that needs to be transmitted has been widely discussed in [3], where both satellite-based and terrestrial (i.e. air-ground) data links are investigated.

For the development of reliable, high-throughput wireless data links, a good understanding of the physical characteristics of the wireless channel between the communication partners, in our case of a terrestrial system the Ground Station (GS) and the UA, is necessary. Wireless channels are characterized by reflection, diffraction, and scattering; these effects are summarized as multi-path propagation [4]. Channel measurements are a common procedure to gain knowledge on wireless channels. This knowledge can later be used for the development and validation of a channel model. The resulting channel model may cover several scenarios – in case of the air-ground channel, those scenarios can be the different phases of a flight like take-off, en-route, and landing. A further differentiation, e.g. based on the ground surface, is also considered in literature, e.g. [5], [6]. The findings from the channel measurements and the resulting channel model can be used to design and evaluate waveforms for a new wireless communication system.

Inherently, channel measurements for aeronautical channels are quite complex and costly as they require at least one air vehicle. Both costs and effort may rise significantly when jet aircraft operating at high speeds at several kilometers of altitude are involved. Additionally, there was no high demand for sophisticated models for aeronautical channels in the past since modern digital communication links did not play a role in civil aviation for decades. These two aspects might be the reasons, why aeronautical channels in general and the aeronautical air-ground channel in particular have not received as much attention as other channels like those used in cellular networks, e.g. LTE and 5 G. However, we want to provide a brief overview of the available literature in the field of aeronautical (air-ground) channel models. A comprehensive review on air-ground channels, with a special focus on UAs can be found in [7].

In [8], [9], the application of simple tap-delay line models on aeronautical channel modeling has been discussed. The authors have already stated, that the different phases of a flight require different models or at least an individual parametrization.

The development of digital data links for civil aviation motivated the development of more sophisticated channel models. For example, a Wide Sense Stationary Uncorrelated Scattering (WSSUS) based channel model for C-band has been developed as part of the adaption of IEEE 802.16e to an airport

Manuscript received 29 July 2022; revised 24 October 2022; accepted 4 December 2022. Date of publication 6 January 2023; date of current version 18 May 2023. The review of this article was coordinated by Prof. Daniel Benevides da Costa. *(Corresponding author: Daniel M. Mielke.)*

The authors are with the German Aerospace Center, Institute of Communications and Navigation, 82234 Wessling, Germany (e-mail: Daniel.Mielke@dlr.de; m.walter@dlr.de; dennis.becker@dlr.de; Miguel.Bellido Manganell@dlr.de; Uwe-Carsten.Fiebig@dlr.de).

Digital Object Identifier 10.1109/TVT.2022.3230357

surface data link called Aeronautical Mobile Airport Communication System (AeroMACS) [10]; however, according to the application of Aeronautical Mobile Airport Communication System (AeroMACS), the model focuses on the airport surface only. An alternative approach for modeling the airport surface channel, also in C-band, has been introduced in [11].

In contrast to AeroMACS, the L-Band Digital Aeronautical Communication System (LDACS) is a communication link designed for all flight phases of manned aircraft, predominantly developed by the German Aerospace Center (DLR). As part of its development, a channel measurement campaign in L-band was performed and a new approach of channel modeling was proposed [12], [13]. The channel model is based on the evaluation of the tracked Multipath Components (MPCs) and does not only incorporate statistical element but also integrates geometric aspects.

In [14], air-ground channel measurements with a bandwidth of 2 MHz in the Ultra High Frequency (UHF) band have been performed. The authors also provide a model on the path loss and discuss the observed effects of multipath propagation by a statistical evaluation of sets of Channel Impulse Responses (CIRs).

Comprehensive channel measurements in both C- and L-band have been performed by a team of the University of South Carolina and NASA. The evaluation has been published in a series of articles: While [5] described the campaign setup and proposed models for the air-ground channel in over-water scenarios, the team focuses on the modeling of the air-ground channel in hilly and mountainous terrain in [6]. In [15], the focus lies on the modeling of the air-ground channel in suburban and near-urban environments. The described models concentrate on the evaluation of sets of CIRs and not on the tracking of individual MPCs over time.

The German Aerospace Center (DLR) also performed a C-band channel measurement campaign with a jet aircraft that covered multiple flight scenarios. This measurement campaign was described in detail in [16], where we have also provided a close analysis of the dominant signal component of the received channel sounding signal. In our present paper we focus on the signal components besides the dominant component, the MPCs, since they, as stated above, strongly define the physical properties of a wireless channel. We address the task of identifying and tracking the MPCs in our measurement data by the application of a multi-stage processing chain. In contrast to [17] we do not apply a filter-based algorithm but a concept, where the MPCs are first identified and are later tracked by a novel path-based algorithm in a second step. This two-staged approach allows a parallelization of the computational expensive processing of the raw measurement data.

The paper is structured as follows: In Section II we briefly describe the setup of the measurement campaign the processed data has been collected. The pre-processing of this data is explained in Section III. Sections IV and V provide information on how the individual MPCs are extracted from the measurement data. In Section VI, we introduce our approach used for the tracking of the detected MPCs over time before we apply it to our measurement data in Section VII. We conclude our paper in Section VIII.

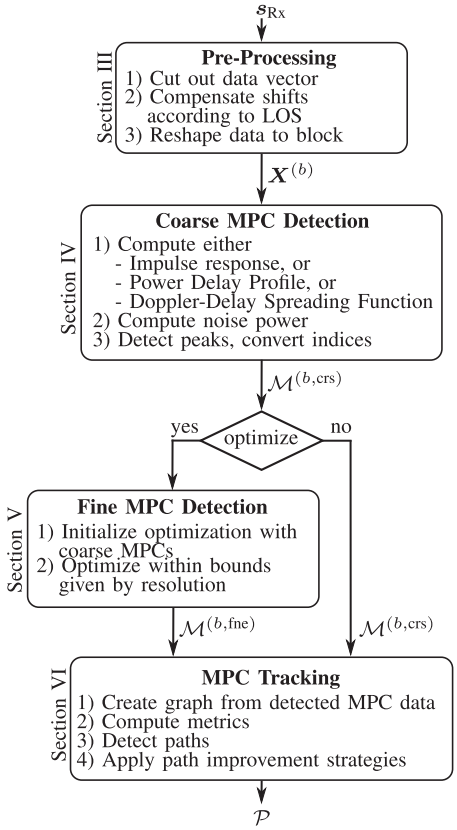

Fig. 1. Block diagram of the processing described in this paper. The section where the corresponding processing step is discussed in detail is given next to the box.

Fig. 1 also provides a graphical overview on how the part of the paper explaining the processing is structured.

II. Measurement Campaign

The DLR performed a measurement campaign in 2018 where the air-ground/ground-air channel in C-band was measured. The applied channel sounding signal had a bandwidth of about 50 MHz and used a carrier frequency of $f_{\mathrm{c}} = 5.2$ GHz. A detailed description of the campaign can be found in [16]; this section just intends to give a brief overview on the campaign setup.

A. Hardware Setup

The campaign setup consisted of a Ground Station (GS) and an Airborne Station (AS) aboard DLR's Falcon 20E aircraft. Through all experiments, the GS was the transmitter of the channel sounding signal and the Airborne Station (AS) was the receiver of the signal.

The most relevant devices of the GS are the Arbitrary Waveform Generator (AWG) and the High Power Amplifier (HPA) that amplifies the output signal of the AWG. The GS also contains a Global Navigation Satellite System (GNSS) receiver and an atomic clock. The latter two devices act as a GNSS-disciplined oscillator that is used as time base for the AWG.

The most important part of the AS setup is the IQ-recorder that samples the received signal, and counts and stores these

IQ-samples. Additionally, the setup contains an Inertial Measurement Unit (IMU) and also a GNSS-disciplined oscillator that is used as a time base for the IQ-recorder. As the applied IQ-recorder is not capable of handling the applied carrier frequency, a frequency mixer ("downconverter") is used to shift the signal to an intermediate frequency. This mixer also provides an Adaptive Gain Control (AGC), that allows an adaptive amplification or attenuation, respectively, of the received signal. It is therefore possible to adjust the receiver's operating point during recording e.g. depending on the Line of Sight (LOS) distance to the transmitter.

B. Measurement Signal and Procedure

The channel sounding signal consists of a so called channel sounding sequence of length 40.96 μs that is gaplessly repeated in an infinite loop on the GS's AWG. The channel sounding sequence is an iteratively filtered multitone signal whose initial phases are distributed as Newman Phases. A more detailed description of the signal generation can be found in Section II-B of [16].

As the receiver samples the incoming signal at $f_{\mathrm{sr}} = 50\,\mathrm{MHz}$, it is assured that $N = 2048$ consecutive samples of the received signal consist exactly one complete instance of the channel sounding sequence.

Before and after all measurement flights, the output of the GS's HPA is directly connected to the receiving hardware aboard the AS using attenuators and a cable to perform a reference measurement of the channel sounding sequence. The transmission and the recording of the channel sounding signal is then started for a few seconds. During post-processing, N consecutive samples of this recorded signal are cut out and called reference signal $\boldsymbol{x}_{\mathrm{ref}} \in \mathbb{C}^N$. We also store the IQ-recorder's sample counter value ρ_{ref} for the first sample of $\boldsymbol{x}_{\mathrm{ref}}$. Additionally, the GNSS-disciplined oscillators of the GS and the AS are synchronized during this process.

During the measurement flight, the GS is in its transmission location on the rooftop of DLR's IKN building and the output of the HPA is connected to the transmitting antenna. The AS's receiving hardware is connected to the receiving antenna mounted at the bottom fuselage of the Falcon aircraft. During the flight, the AS's IQ-recorder stores the received samples in $\boldsymbol{s}_{\mathrm{Rx}}$.

It is ensured, that the GS's AWG and the AS's IQ-recorder are continuously running during and between the pre-flight reference measurement, the actual measurement flight, and the post-flight reference measurement. Thus, we can simply use the sample counter ρ of the IQ-recorder as a reliable time base during offline processing, when the measurement data recorded during flight are evaluated using the reference signal $\boldsymbol{x}_{\mathrm{ref}}$, see Sections III to V.

C. Flight Tracks and Maneuvers

All flights started and ended at the EDMO airport close to Munich, Germany. The campaign involved four flights where different flight scenarios were covered. The flight scenarios not only included typical en-route flight patterns at cruising speed at multiple cruising altitudes, but also take-offs, landings, go-arounds, and flights with extreme banking. The data collected during the different flight scenarios allows us to develop individual channel models for each scenario.

III. Pre-Processing

A. Ground Truth

Based on the data recorded by both GNSS-receivers and the IMU, a ground truth is computed. The ground truth provides information on the aircraft's position, orientation, and heading at a certain time instant. This allows us to estimate the LOS component's Free Space Path Loss (FSPL), the LOS component's delay, denoted by τ_{LOS}, and the LOS component's Doppler shift, denoted by ν_{LOS}.

B. Block Processing

The processing of the measurement data is performed block wise. A block b of BN consecutive samples is cut out of the received signal:

$$\begin{aligned}\boldsymbol{x}^{(b)'} &= \boldsymbol{s}_{\mathrm{Rx}}[\rho^{(b)}, \ldots, \rho^{(b)} + BN] \\ &= \boldsymbol{s}_{\mathrm{Rx}}[\rho^{(b)}, \ldots, \rho^{(b+1)} - 1],\end{aligned} \tag{1}$$

where $\rho^{(b)}$ denotes the IQ-recorder's sample counter corresponding to block b. We call $B \in \mathbb{N}$ the *block size*, as it defines how many consecutive channel sounding sequence instances of length N are processed together in one block. A larger B increases the Doppler resolution during processing, however, this increase comes at the cost of time resolution: If B is chosen large, it is possible that relevant fast fading channel effects are vanished out and thus are not modeled appropriately later. This effect becomes even more likely considering the comparatively high velocity of an aircraft.

Based on the ground truth data, $\boldsymbol{x}^{(b)'}$ is shifted such that the current LOS delay and LOS Doppler shift are compensated:

$$\boldsymbol{x}^{(b)} = \mathfrak{F}_{-\tau_{\mathrm{LOS}}^{(b)}}^{-\nu_{\mathrm{LOS}}^{(b)}} \left\{ \boldsymbol{x}^{(b)'} \right\}, \tag{2}$$

where $\mathfrak{F}_{\tau}^{\nu}\{\cdot\}$ denotes a function shifting a signal by delay τ and frequency ν.

In a last step, we reshape the vector $\boldsymbol{x}^{(b)} \in \mathbb{C}^{BN}$ to a matrix $\boldsymbol{X}^{(b)} \in \mathbb{C}^{N \times B}$, where each column corresponds to one instance of the channel sounding sequence, thus the items of $\boldsymbol{X}^{(b)}$ are set according to:

$$\begin{aligned}\boldsymbol{X}^{(b)}[n, m] = \boldsymbol{x}^{(b)}[mB + n], \\ \forall\, n \in \{0, \ldots, N-1\}, \\ \forall\, m \in \{0, \ldots, B-1\}.\end{aligned} \tag{3}$$

In the following, we assume that the power levels of $\boldsymbol{x}_{\mathrm{ref}}$ and $\boldsymbol{x}^{(b)}$ are matched, such that the attenuation used during the recording of the reference signal and the FSPL of the LOS component during the recording of $\boldsymbol{x}^{(b)}$ are compensated as suggested in [16, (11)].

IV. Coarse Multipath Component Detection

In this section, we describe our approach for detecting Multipath Components (MPCs) in our measurement data. As shown in [17], an MPC l can be described by a quadruple

$$\xi_l = (\ \tau_l,\ \nu_l,\ |\alpha_l|\ ,\ \arg\{\alpha_l\}\ )\ , \tag{4}$$

where τ_l denotes the delay shift, ν_l the Doppler shift, and α_l the complex weight of the MPC.[1] The weight α_l is split into its absolute value and its phase to achieve $\xi_l \in \mathbb{R}^4$, which not only allows a direct access to the MPCs' amplitudes, but also a more memory efficient processing. If not denoted otherwise, we assume that both τ_l and ν_l are given with respect to the LOS component.

The set of MPCs that have been detected in a processing block b is denoted by $\mathcal{M}^{(b)}$ having cardinality $L^{(b)}$. To avoid ambiguities, the block index b is added to the MPC notation according to $\xi_l^{(b)}$ if necessary.

We now describe the processing steps for the *coarse* detection of MPCs or clusters of MPCs, respectively. The detected set of MPCs is therefore denoted by $\mathcal{M}^{(b,\mathrm{crs})}$.

Please note that the described approaches have no data dependencies on previously processed blocks, which allows a parallel processing of an arbitrary amount of blocks at the same time.

A. Based on Impulse Response/Power Delay Profile

The basic idea of this approach is to detect peaks in the absolute value of the Impulse Response (IR) or the Power Delay Profile (PDP), respectively, of a data block. Each of these peaks represents either a single MPC or a cluster of MPCs; the position of the peaks can be used to estimate the underlying MPCs' delay τ_l. However, no Doppler information ν_l can be extracted (directly) following this approach.

1) Impulse Response/Power Delay Profile: The first step is to compute either the Impulse Responses (IRs) or the complex Power Delay Profile (PDP) of the current data block $\boldsymbol{X}^{(b)}$ by applying the processing described in Appendix A or Appendix B, respectively. While the (coherent) PDP has the advantage of a lower noise floor, short term MPCs may vanish out if the block size B is chosen too large. On the other hand, if B is chosen too small, signals of some reflectors may vanish in the noise floor and will remain undetected.

As the further processing is the same for both cases, in the following, we refer to both the individual IRs of the current block and the PDP, respectively, by $\boldsymbol{y}^{(b)} \in \mathbb{C}^{N f_{\mathrm{up}}}$. f_{up} denotes the upsampling factor.

We also define the corresponding logarithmic vector

$$\boldsymbol{y}_{\log}^{(b)} = 10\log_{10}\left|\boldsymbol{y}^{(b)}\right|. \tag{5}$$

2) Noise Floor: The power of the noise floor for the current block ${\sigma^{(b)}}^2$ is estimated based on areas of $\boldsymbol{y}_{\log}^{(b)}$ where the appearance of measurable strong MPCs or measurement artifacts is very unlikely. Thus, the average power for the part of $\boldsymbol{y}_{\log}^{(b)}$ where $\{\tau_\sigma | \tau_{\max} < \tau_\sigma < N/f_{\mathrm{sr}}\}$ is computed, where $\tau_{\max}$ denotes the threshold delay up to which significant MPCs are expected.

3) Peak Detection: The next task is to detect local maxima ("peaks") in $\boldsymbol{y}_{\log}^{(b)}$. A peak must fulfill the following two conditions:

- The peak's value must exceed a value of ${\sigma^{(b)}}^2|_{\mathrm{dB}} + P_{\mathrm{thresh}}$.
- The peak must have a certain prominence defined by a minimum distance to other detected peaks.

All detected $L^{(b)}$ peaks' positions n_l inside of $\boldsymbol{y}_{\log}^{(b)}$ are stored in $\mathcal{P}^{(b)}$, where n_l denotes the index of the l-th peak.

4) Index Translation: Based on the resolution of the IRs given by (22), the indices found in the previous step can be translated to a delay according to

$$\tau_l = n_l \Delta_\tau. \tag{6}$$

The complex weight is given by $\alpha_l = \boldsymbol{y}^{(b)}[n_l]$. In order to complete the four elements of the MPC as defined in (4), the Doppler shift of MPC l is set to $\nu_l = 0$. Then, ξ_l is added to $\mathcal{M}^{(b,\mathrm{crs})}$.

B. Based on Doppler-Delay Spreading Function

The basic idea of this approach is very similar to the previous one, however, instead of using the IR/PDP, the Doppler-Delay Spreading Function (DDSF) is used. The Doppler-Delay Spreading Function (DDSF) provides not only information on the delay, but also on the Doppler shift of an MPC. Again, peaks in the DDSF represent either a single MPC or a cluster of MPCs.

1) Doppler-Delay Spreading Function: The detection process starts with the computation of the DDSF according to Appendix C of the current data block b as defined above and take its logarithmic absolute value:

$$\boldsymbol{A}^{(b)} = 10\log_{10}\left|\mathcal{S}_{\boldsymbol{x}_{\mathrm{ref}}}^{f_{\mathrm{up}}}\left\{\boldsymbol{X}^{(b)}\right\}\right|, \tag{7}$$

where f_{up} denotes the upsampling factor along the delay axis i.e. the columns of $\boldsymbol{A}^{(b)}$; the rows correspond to the Doppler axis.

2) Noise Floor: The power of the noise floor for the current block ${\sigma^{(b)}}^2$ is estimated based on areas of $\boldsymbol{A}^{(b)}$ where the appearance of measurable strong MPCs or measurement artifacts is very unlikely. Thus, the average power for the areas where $\{\tau_\sigma | \tau_{\max} < \tau_\sigma < N/f_{\mathrm{sr}}\}$ and $\{\nu_\sigma | \nu_{\min} < |\nu_\sigma| < \nu_{\max}\}$ is computed.

3) Peak Detection: The next task is to detect local maxima ("peaks") in $\boldsymbol{A}^{(b)}$. A peak must fulfill the following two conditions:

- The peak's value must exceed a value of ${\sigma^{(b)}}^2|_{\mathrm{dB}} + P_{\mathrm{thresh}}$.
- The peak must have a certain prominence defined by an elliptical footprint. This allows a more flexible search than a simple distance measure.

All detected $L^{(b)}$ peaks' coordinates inside of $\boldsymbol{A}^{(b)}$ are stored pairwise (n_l, m_l) in $\mathcal{P}^{(b)}$, where n_l denotes the row index and m_l denotes the column index of the l-th peak.

4) Index Translation: Based on the resolution of the DDSF given by (22), and (23), the index pairs found in the previous step can be translated to a delay and Doppler value. For the delay, this

[1] Note that MPCs that cannot be resolved due to limited resolution are handled as a single MPC, although the more correct term would be *MPC cluster*.

mapping is done according to (6), for the Doppler, the mapping is done according to

$$\nu_l = \left(m_l - \frac{B}{2}\right)\Delta_\nu. \tag{8}$$

5) Least Squares Optimization: To find the complex weights for the MPCs, we first create a matrix describing a synthesized version of the signal based on the delays and Doppler shifts of the MPCs detected above using the shift function $\mathfrak{F}_\tau^\nu\{.\}$ introduced in Section III:

$$\boldsymbol{Y}^{(b)} = \begin{pmatrix} \mathfrak{F}_{\tau_0}^{\nu_0}\left\{\boldsymbol{x}_{\text{ref}}^{(B)}\right\} \\ \mathfrak{F}_{\tau_1}^{\nu_1}\left\{\boldsymbol{x}_{\text{ref}}^{(B)}\right\} \\ \vdots \\ \mathfrak{F}_{\tau_{L^{(b)}-1}}^{\nu_{L^{(b)}-1}}\left\{\boldsymbol{x}_{\text{ref}}^{(B)}\right\} \end{pmatrix}, \quad \boldsymbol{Y}^{(b)} \in \mathbb{C}^{L^{(b)}\times BN}, \tag{9}$$

where $\boldsymbol{x}_{\text{ref}}^{(B)}$ denotes a vector with B concatenated instances of the reference signal. Thus, the l-th row of $\boldsymbol{Y}^{(b)}$ contains a modified version of the reference signal, shifted in time and frequency according to τ_l and ν_l, respectively.

Now, an approximation for the vector $\boldsymbol{\alpha}^{(b)} \in \mathbb{C}^{L^{(b)}}$ needs to be found for the equation

$$\boldsymbol{Y}^{(b)^T}\boldsymbol{\alpha}^{(b)} = \boldsymbol{x}^{(b)}. \tag{10}$$

We apply the Least Squares (LS) algorithm for this task. Besides the desired approximation for $\boldsymbol{\alpha}^{(b)}$, the Least Squares (LS) algorithm also returns a residual $\epsilon^{(b)}$ that we understand as a measure of precision of the estimated MPCs' parameters. The residual is computed according to

$$\epsilon^{(b)} = \sum_{BN} \underbrace{||\boldsymbol{x}^{(b)} - \boldsymbol{Y}^{(b)^T}\boldsymbol{\alpha}^{(b)}||^2}_{\in\mathbb{R}^{BN}}, \tag{11}$$

which corresponds to the sum over all elements of the vector that results from the squared Euclidean distance between the measured signal block and the weighted synthesized signal.

By combining τ_l, ν_l, and the absolute value and phase of the l-th entry of $\boldsymbol{\alpha}^{(b)}$, all elements of MPC ξ_l, as defined in (4), are given. ξ_l is now added to $\mathcal{M}^{(b,\text{crs})}$.

V. Detection Improvements

A. Fine Detection

The potential problem of the approach described in Section IV-B is, that its precision is limited by the resolution of the discrete DDSF. Therefore, the exact position of an MPC inside of a tile of the DDSF matrix – and consequently its exact delay and Doppler shift – cannot be determined any further.

We address the this issue by applying an optimization algorithm that tries to adjust the delay and Doppler shift of every detected MPC within the bounds defined by the size of a tile of the DDSF.

1) Optimization Algorithm: As suggested in [17], we apply the Bound Optimization by Quadratic Approximation (BOBYQA) algorithm [18] to the given problem. The BOBYQA algorithm can be applied to multi-dimensional problems, does not require a derivative of the optimization objective, and allows the usage of optimization bounds.

2) Initialization: The optimization is initialized by the delay and the Doppler shifts of the MPCs given in $\mathcal{M}^{(b,\text{crs})}$.

3) Dimensions: The optimization algorithm tries to achieve an optimal result by adjusting the delay *and* Doppler shift of every MPC. As the amount of assumed MPCs does not change during the processing of block b, the optimization problem is solved along $2|\mathcal{M}^{(b,\text{crs})}| = 2L^{(b,\text{crs})}$ dimensions.

4) Optimization Bounds: The optimization bounds for each MPC are given by the DDSF's tiles' dimensions. Thus, the lower bound (lb) and the upper bound (ub) for the delay optimization of MPC l are given by:

$$\tau_l^{(\{\text{lb,ub}\})} = \max\left\{0,\ \tau_l \mp \frac{\Delta_\tau}{2}\right\} \tag{12}$$

and, correspondingly, for the Doppler optimization of MPC l, the bounds are given by

$$\nu_l^{(\{\text{lb,ub}\})} = \nu_l \mp \frac{\Delta_\nu}{2}. \tag{13}$$

5) Optimization Process: The algorithm basically executes the LS optimization described in Section IV-B5 in an iterative loop indexed by $i \in \mathbb{N}$:

First, a matrix $\tilde{\boldsymbol{Y}}_i^{(b)}$ is generated using the delay and Doppler shifts of the MPCs given in the current (hypothetical) set of MPCs $\tilde{\mathcal{M}}_i^{(b)}$ similar to (9). Then, an equation similar to (10)

$$\tilde{\boldsymbol{Y}}_i^{(b)T}\tilde{\boldsymbol{\alpha}}_i^{(b)} = \boldsymbol{x}^{(b)} \tag{14}$$

is solved for $\tilde{\boldsymbol{\alpha}}_i^{(b)}$ – again using the LS approach.

Thus, the overall objective of the optimization problem is to minimize the residual $\epsilon_i^{(b)}$ returned by the LS algorithm applied to (14). Using the definition given in (11), this leads to

$$\mathcal{M}^{(b,\text{fne})} = \arg\left\{\min_{\tilde{\mathcal{M}}_i^{(b)}}\left\{\underbrace{\sum_{BN}\left|\left|\boldsymbol{x}^{(b)} - \tilde{\boldsymbol{Y}}_i^{(b)T}\tilde{\boldsymbol{\alpha}}_i^{(b)}\right|\right|^2}_{\epsilon_i^{(b)}}\right\}\right. \tag{15}$$

6) Termination: The execution of the optimization is terminated when the result does not improve more than a certain threshold from one iteration to the next: $\epsilon_{i-1}^{(b)} - \epsilon_i^{(b)} < \epsilon_{\text{thresh}}$.

B. Cluster Resolution

Although the approach presented in Section V-A has the ability to determine the parameters of the MPCs even below the resolution of the DDSF, it cannot resolve multiple MPCs whose mutual distances in delay and/or Doppler are lower than the corresponding DDSF resolution. Thus, they can only be described as clusters and not get detected individually. In case this is not acceptable, we suggest the following approach to resolve the individual MPCs.

It is assumed that the coarse detection process as defined in Section IV-B has been completed; thus there is a set $\mathcal{M}^{(b,\text{crs})}$. For

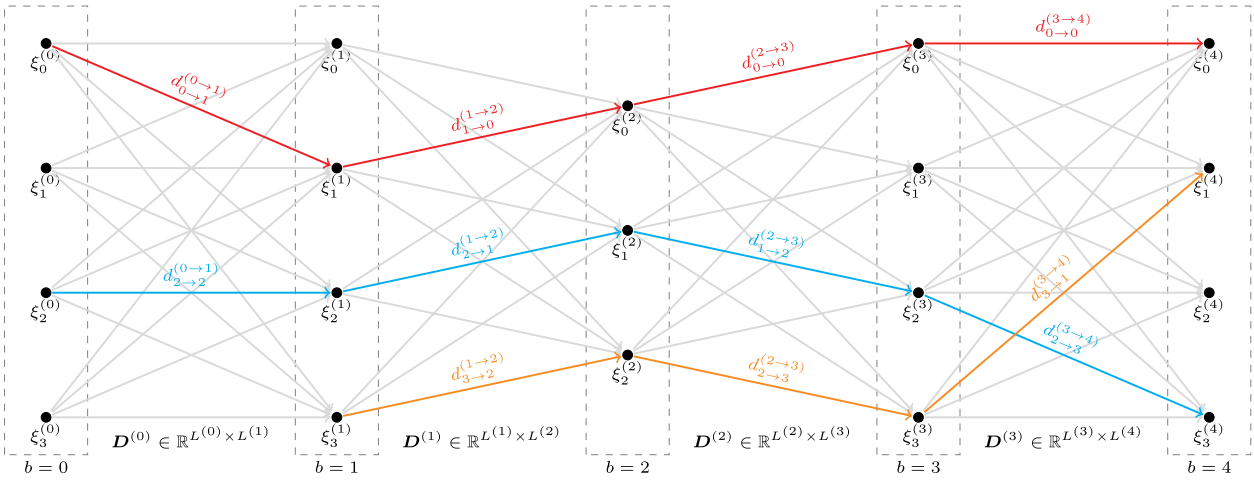

Fig. 2. Graph representation of the MPCs $\xi_l^{(b)}$ detected in five consecutive blocks indexed by b; MPCs are represented by a black dot, blocks are represented by a dashed-line rectangle: While only three MPCs have been detected in block $b = 2$, four MPCs have been detected in the other blocks. The arrows connecting the MPCs represent the distance metrics $d_{l_\mathrm{s} \to l_\mathrm{e}}^{(b \to b+1)}$ between all MPCs of consecutive blocks; however, for the sake of clearness, only few distance metrics are given in the figure. All distance metrics between block b and block $b + 1$ are given in $\boldsymbol{D}^{(b)}$.

each item in $\mathcal{M}^{(b,\mathrm{crs})}$, $\Psi \in \mathbb{N}$ copies are created. A random delay and Doppler shift is added to each of these copies according to

$$\forall\, l \in \{0, 1, \ldots, L^{(b,\mathrm{crs})} - 1\} :$$
$$\forall\, \psi \in \{0, 1, \ldots, \Psi - 1\} :$$
$$\tau_{l,\psi} = \tau_l + \left(\mathcal{U}_{[0,1]} - 0.5\right) \Delta_\tau \tag{16}$$
$$\nu_{l,\psi} = \nu_l + \left(\mathcal{U}_{[0,1]} - 0.5\right) \Delta_\nu, \tag{17}$$

where τ_l and ν_l correspond to the delay and Doppler shift of $\xi_l \in \mathcal{M}^{(b,\mathrm{crs})}$ and $\mathcal{U}_{[v,w]}$ denotes a uniform distribution on the interval $[v, w]$. These newly created MPCs are denoted by $\xi_{l,\psi}$ and summarized in $\mathcal{M}_\Psi^{(b,\mathrm{crs})}$.

So far, this approach alleges, that *every* item in $\mathcal{M}^{(b,\mathrm{crs})}$ is in fact a MPC cluster. However, the idea is that due to the further processing, i.e. by applying the optimization algorithm from Section V-A, the weights of the "unnecessary" MPCs copies will tend to 0:

The algorithm given in Section V-A is initialized using $\mathcal{M}_\Psi^{(b,\mathrm{crs})}$. For each $\xi_{l,\psi}$, the optimization bounds are the same as for the corresponding "parent" MPC ξ_l.

The amount of optimization dimensions is $2\Psi L^{(b,\mathrm{crs})}$. Once the termination criterion is fulfilled (see above), the resulting MPCs are given in $\mathcal{M}_\Psi^{(b,\mathrm{fne})}$.

VI. Tracking

Tracking of MPCs describes the behavior of the components over time. This is often realized by filter-based approaches, e.g. in [17]. However, path-based approaches have also been discussed, e.g. in [19]. We now describe our path based approach to track MPCs over time, i.e. multiple consecutive blocks b.

A. Graph Representation

Based on the sets of MPCs $\mathcal{M}^{(b)} \forall\, b$ detected using one of the approaches given in Sections IV and V, a directed graph $G = (\mathcal{M}, \mathcal{D})$ as given in Fig. 2 is created. The nodes $\mathcal{M}$ of the graph are given by the union of all MPC sets: $\mathcal{M} = \bigcup_b \mathcal{M}^{(b)}$. The edges $\mathcal{D}$ of the graph are given by the set of all distances between the MPCs of consecutive blocks. These distances are defined as follows: The distance from the l_s-th MPC (start) of the b-th block to the l_e-th MPC (end) of the $(b + 1)$-th block is given by

$$d_{l_\mathrm{s} \to l_\mathrm{e}}^{(b \to b+1)} = \mathfrak{D}\left\{\xi_{l_\mathrm{s}}^{(b)}, \xi_{l_\mathrm{e}}^{(b+1)}\right\}, \tag{18}$$

where $\mathfrak{D} : \mathbb{R}^4 \times \mathbb{R}^4 \mapsto \mathbb{R}$ is a function expressing the distance between two MPCs by a real value.

The entry of the l_s-th row and the l_e-th column of the distance matrix $\boldsymbol{D}^{(b)} \in \mathbb{R}^{L^{(b)} \times L^{(b+1)}}$ equals $d_{l_\mathrm{s} \to l_\mathrm{e}}^{(b \to b+1)}$.

We furthermore assume, that all MPCs of block b are sorted in descending order based on their absolute weight: $|\alpha_l^{(b)}| \geq |\alpha_{l+1}^{(b)}|$.

B. Path Detection

We use a path, represented by a sequence $\gamma = ((b, l)_i)_{i=0,1,\ldots,\beta-1}$, to describe the behavior of an MPC over time. Each element of a path is a tuple (b, l) that sufficiently describes the position of each node inside of G. To avoid ambiguities, we add a superscript to our notation (e.g. $\gamma^{(a)}$) when necessary. The set of all paths is denoted by $\mathcal{P}$.

For the sake of simplification, we have split the description of our path detection algorithm into three parts: the initialization, an *inner* part, and an *outer* part.

1) Initialization: For each block b, a vector $\Gamma^{(b)} \in \mathbb{N}^{L_b}$ is defined and all of its items are initially set to zero. The purpose of this vector is to indicate whether an MPC has been assigned to a path: If MPC l of block b gets assigned to a path, $\Gamma^{(b)}[l] = 1$ is set.

A threshold $d_\mathrm{thresh} \in \mathbb{R}$ is defined that provides a maximum distance two MPCs are allowed to have within a path. If this

Listing 1: Pseudo code of the outer part of the path detection algorithm. See Listing 2 for the *detect_path* subroutine.

```
1 # ordered list where all detected paths are stored
2 list all_paths = ()
3
4 # iterate over all blocks along time
5 for int b in range(0, num_blocks-1):
6
7     # run until all MPCs of current block are assigned
          to a path
8     while not all(Γ^(b) == 1):
9
10        # iterate over MPCs of current block
11        for int l in range(0, L^(b)-1):
12
13            # check if MPC is already assigned to a path
14            if Γ^(b)[l] == 1:
15
16                # it is → continue with next MPC
17                continue
18
19            # call subprocedure to detect path
20            list γ = detect_path(
21                root_block_index = b,
22                root_mpc_index = l
23            )
24
25            # append path to path list
26            all_paths.append(
27                γ
28            )
```

Listing 2: Pseudo code of the detect path subroutine called by Listing 1.

```
1  sub detect_path(
2        int root_block_index,
3        int root_mpc_index
4    ) → list:
5
6    # initialize ordered list for new path
7    list γ = ()
8
9    # add tuple defining path root
10   γ.append(
11       (root_block_index, root_mpc_index)
12   )
13
14   # init MPC index for loop
15   int l = root_mpc_index
16
17   # iterate over all subsequent blocks
18   for int b in range(root_block_index, num_blocks-2):
19
20       # get list of unassigned MPCs in next block
21       list avail_indices = index( Γ^(b+1) == 0 )
22
23       # check if all MPCs are already assigned
24       if empty(avail_indices):
25
26           # they are → break loop
27           break
28
29       # find minimum distance by applying (18)
30       float d_min = min_{∀ l̃ ∈ avail_indices} { 𝔇 { ξ_l^(b), ξ_l̃^(b+1) } }
31
32       # check if min. distance is above threshold
33       if d_min > d_thresh:
34
35           # it is → break loop
36           break
37
38       # get index of MPC with min. distance and use
             it as new MPC index for next iteration
39       int l = arg{d_min}
40
41       # add detected path segment
42       γ.append(
43           (b+1, l)
44       )
45
46       # mark MPC as assigned
47       Γ^(b+1)[l] = 1
48
49   # return the list describing the path
50   return γ
```

threshold is exceeded, the respective MPC cannot be part of the given path.

For the sake of completeness, an ordered list (corresponding to $\mathcal{P}$) is created in Listings 1, line 2 where all detected paths are stored.

The purpose of the outer part (see Listing 1) of the algorithm is to detect the root note of a new path. Once a root element has been found, the inner part, represented by the subroutine *detect_path*, is called to find the remaining elements of the path. In the following, we explain the outer part of the algorithm in detail:

The loop defined in line 5 iterates over the blocks starting at the first block. Considering the graph in Fig. 2, this loop iterates over the blocks from left to right.

The loop in line 8 ensures that the algorithm keeps processing on the current block b, until all of its MPCs have been assigned to a path.

The loop defined in line 11, finally, iterates over the MPCs of the current block b. Unless the current MPC l has been assigned to a path yet (check in line 14), the actual path detection subroutine *detect_path*, corresponding to the inner part of our approach, is called in line 20. The arguments passed to this call are the current block b and MPC l as they are used as the root of the new path.

Once *detect_path* returns a new path γ, it is appended to the global path pool in line 26.

Please note that the update of $\Gamma^{(b)}$ is performed inside of the called subroutine. Due to this side-effect we do not call *detect_path* a function to highlight the lack of idempotence.

3) Inner Part: The inner part of the algorithm is given in the subroutine *detect_path* in Listing 2. As the arguments passed to this subroutine describe the root of a new path inside of the graph, this tuple is added as a first element (line 10) to the list representing the path γ.

During this subroutine, the MPC index of the last element of the path is represented by l; in the beginning, this corresponds to the MPC index of the root node (line 15).

Similar to the block loop in Listing 1, the block loop of the subroutine iterates from "left" ro "right" over the blocks, starting with the root block (line 18).

Within the loop, the distances from the current MPC l to all MPCs of the next block that are not yet assigned to a path are computed and the minimum is determined (line 30). Please note, that the actual call of $\mathfrak{D}$ can be substituted by a lookup in the corresponding distance matrix $\boldsymbol{D}^{(b)}$.

In case either no unassigned MPCs can be found (check in line 24) or the determined minimum distance exceeds the threshold d_{thresh} (check in line 33), the loop stops – corresponding to a termination of the current path.

If none of these conditions is fulfilled, the index of the MPC with the minimum distance is assigned to l (line 39) and the new node is appended to the path γ (line 42). Finally, the detected MPC is marked as assigned (line 47).

Once the loop has terminated, no matter what criterion caused the loop's termination, the subroutine returns the detected path γ.

C. Path Improvements

The algorithm presented in Section VI-B has properties that may result in a degraded MPC tracking under certain conditions. We suggest the following approaches to address these issues.

1) Path Merging: In case an MPC remains undetected for the duration of a single (or more) block(s), e.g. because it does not exceed the detection threshold, an actual path is detected as multiple individual paths. To detect and connect these path segments, we evaluate the distances between the last element of all paths and the root elements of all other paths:

We first set a threshold $\beta_{\max} \in \mathbb{N}$ that defines the maximum length of interruptions given in blocks that we consider as acceptable. We then iterate over all permutations of all paths $\gamma^{(m)}$ and $\gamma^{(n)}$ to detect all path combinations, where

$$\begin{aligned} 0 < b_{\text{root}}^{(n)} - b_{\text{end}}^{(m)} < \beta_{\max}, \text{ with} \\ (b_{\text{root}}^{(n)}, l_{\text{root}}^{(n)}) := (b,l)_{0}^{(n)} \in \gamma^{(n)} \\ (b_{\text{end}}^{(m)}, l_{\text{end}}^{(m)}) := (b,l)_{\beta^{(m)}-1}^{(m)} \in \gamma^{(m)} \end{aligned} \tag{19}$$

holds. The paths $\gamma^{(m)}$ and $\gamma^{(n)}$ are merged if

$$d_{l_{\text{end}}^{(m)} \to l_{\text{root}}^{(n)}}^{(b_{\text{end}}^{(m)} \to b_{\text{root}}^{(n)})} < d_{\text{thresh}} \tag{20}$$

is fulfilled. In case this condition holds for multiple path pairs, the one with the lowest distance is chosen.

So far, the stability of a path $\gamma^{(m)}$ was equal to its length $\beta^{(m)}$. However, the process of path merging motivates the definition of a new measure of the path stability, as the path now may skip some blocks. We therefore define the path stability $\eta^{(m)}$ as the delta of the block index of the last element of the path end the block index of the root element of a path: $\eta^{(m)} = b_{\text{end}}^{(m)} - b_{\text{root}}^{(m)}$.

2) Short Path Elimination: MPCs that appear for the duration of just one block are likely to be the results of a mis-detection. In this case, the algorithm in Section VI-B assigns these single MPCs to individual paths of length one. To only keep track of MPCs that have been detected for a longer duration, a threshold $\beta_{\min} \in \mathbb{N}$ can be set, that defines a minimum length of paths. Consequently, all paths that do not fulfill

$$\eta^{(m)} \geq \beta_{\min} \tag{21}$$

are then eliminated.

If the proposed MPC tracking approach is used for channel modeling, those MPCs that have been assigned to distinct paths which have not been eliminated can be modeled by a process that generates paths. However, the MPCs assigned to one of the eliminated paths should not be ignored: We interpret these MPCs as a result of diffuse scattering and suggest to model them by a separate process. Thus, $\beta_{\min}$ is understood as a parameter that has an influence on the amount of MPCs that are modeled by the process that generates paths. The higher this amount, the smaller the amount of MPCs that are modeled by the separate process.

VII. Results and Discussion

In this section we present some results from the different processing steps described above. Due to the power normalization assumed in Section III, all amplitudes are given in dB with respect to the transmitted power and compensated FSPL. Delays and Doppler shifts are given with respect to LOS if not denoted otherwise.

A. Detection

1) Take-Off: Fig. 3 shows a color-coded representation of the DDSFs of three consecutive blocks $b \in \{0,1,2\}$ computed according to the steps described in Section IV-B. The underlying data were recorded during take-off and processed with a block size of $B = 480$, corresponding to a block duration of $BN/f_{\text{sr}} = 19.6\,\text{ms}$. The detected MPCs are highlighted by red markers.

All three figures show two main reflector areas: One that corresponds to the LOS path and its close environment and one that consists of multiple MPCs with a Doppler shift of around $-1.1\,\text{kHz}$ and a delay of $2.1\,\mu\text{s}$ to $3.5\,\mu\text{s}$. A third, less intense reflector area can be seen for a Doppler shift of around $0.8\,\text{kHz}$ and a delay of $4.0\,\mu\text{s}$ to $4.3\,\mu\text{s}$. Only very few MPCs with a positive Doppler shift can be observed.

It can be observed, that most of the MPCs detected in block $b = 0$ are also detected for $b = 1$ and $b = 2$ and vice versa. However, this does not apply for some MPC: Especially those having a delay of more than $5.8\,\mu\text{s}$ tend to be missed in some blocks during detection. At the same time, it can be observed that these MPCs not only have a comparatively huge delay, but are also comparatively weak in power. This behavior motivates the concept of *Path Merging* (see Section VI-C) during the MPC tracking process.

2) Flyover: Fig. 4 also shows a color-coded representation of the DDSFs of three consecutive blocks $b \in \{0,1,2\}$. The underlying data were recorded just seconds before a direct flyover of the AS over the GS in an altitude of around 3.2 km. All processing parameters were the same as for the take-off scenario described above.

First, it can be observed that the noise floor is about 7.3 dB stronger compared to the take-off scenario. We explain this by the applied AGC setting of the mixer, see Section II-A, during recording: As the LOS distance – and consequently the FSPL – was higher than during the recording of the data shown in Fig. 3, a higher pre-amplification of the received signal was required, including an increase of the noise floor.

It can also be observed, that the LOS component is blurred along the Doppler axis. We explain this by the fact, that the aircraft was moving, while the received data are processed in blocks. Theoretically, block processing as applied here requires a perfect snapshot of the received data where *all* parameters that affect the measurement remain *exactly* the same for the length of one block. For obvious reasons this is not possible in a real world scenario, especially when an aircraft is moving at cruising speed. However, to extract any kind of Doppler information from measurement data, some sort of block processing is required as the Doppler can be described as the time derivative of the delay.

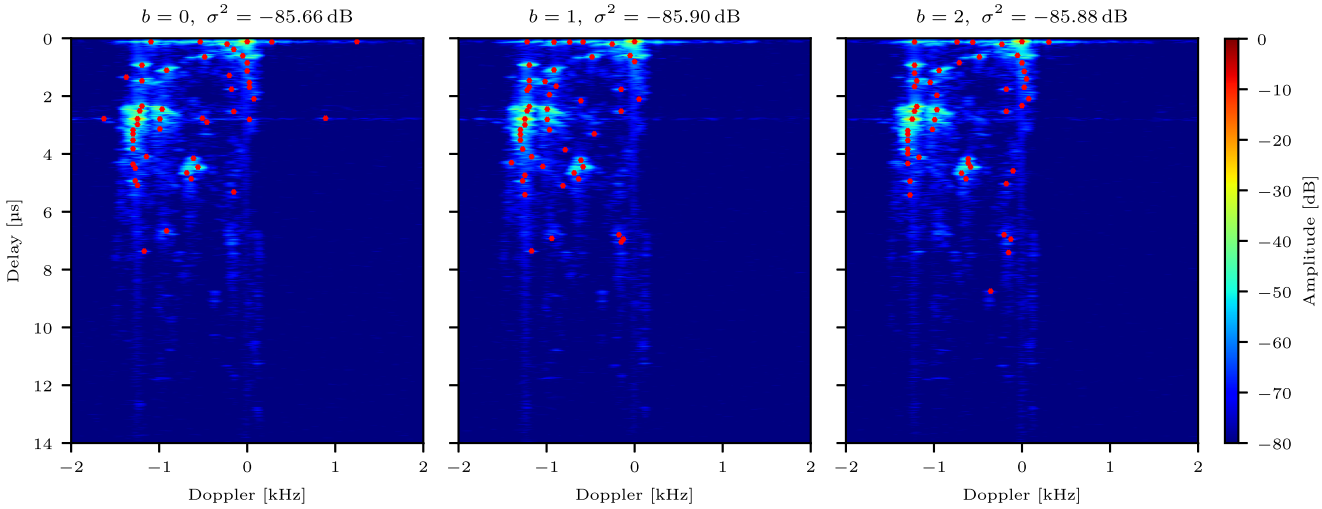

Fig. 3. Color-coded representation of the logarithmic DDSFs of three consecutive blocks recorded during take-off. The strongest 50 MPCs that also exceed a power 15 dB above the respective noise floor σ^2 are highlighted by red crosses. All blocks have been processed individually without any time dependent data dependency.

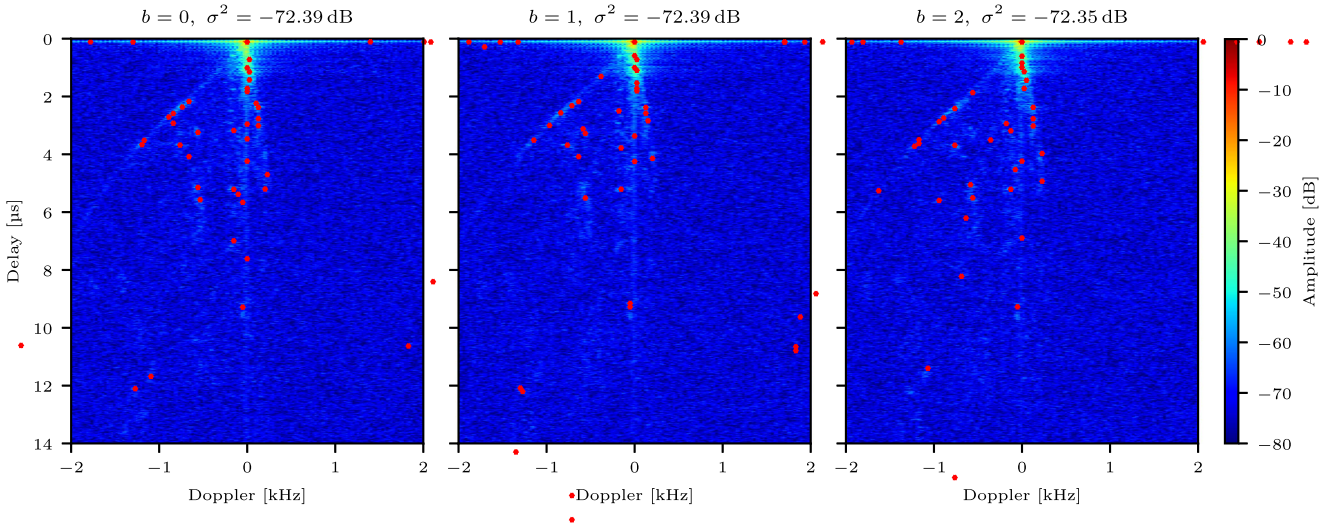

Fig. 4. Color-coded representation of the logarithmic DDSFs of three consecutive blocks recorded during a flyover over the GS. The strongest 50 MPCs that also exceed a power 8 dB above the respective noise floor σ^2 are highlighted by red crosses. All blocks have been processed individually without any time dependent data dependency.

Thus, a trade-off between available Doppler resolution and time precision has to be resolved.

The envelope of the detected MPCs shows a shape similar to a parabola. This is not a coincidence, but can be explained by the fact, that in the air-ground channel, most of the reflectors causing the MPCs are located on the ground comparatively close to the GS. At the same time, each reflector is located on the surface of a prolate spheroid having the transmitter and receiver in its focal points [20]. The size of the spheroid a given reflector is located on is determined by the corresponding MPC's delay. As most of the reflectors that cause the MPCs are distributed on the earth's surface, which can be modeled by a plane, the intersection of the prolate spheroid and the surface results in a parabola-like shape.

B. Path-Based MPC Tracking

Fig. 5 shows a 5 s cutout of the evolution of the MPC components represented as paths. The MPCs were detected using the approach shown in Section IV-B; the paths were found using the algorithm presented in Section VI using the delay-Doppler-only metric as defined in Appendix D. The underlying data were recorded while the aircraft was on the runway and accelerating for take-off. For the sake of clearness, short paths having a length less than 50 blocks have been excluded from the plots.

1) Interpolated Data Points: Fig. 5(a) shows the paths as they were detected by the algorithm including the application of the path improvements presented in Section VI-C. Due to the *path*

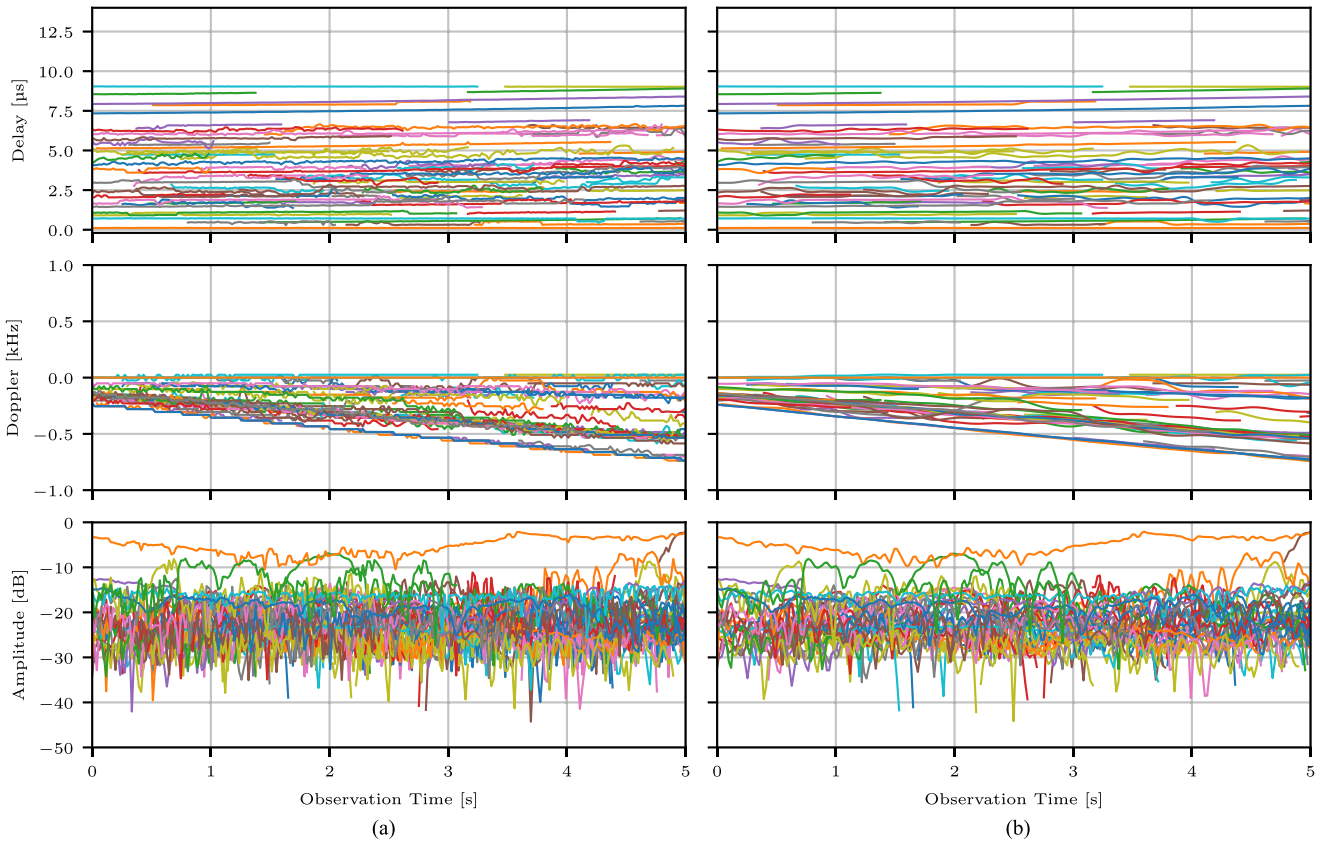

Fig. 5. A set of MPC-paths showing the evolution of delay (top), Doppler (middle), and the absolute value of the weight (bottom) of the detected MPCs during take-off. In both a) and b) missing values were interpolated using a cubic spline interpolation where possible. In b), an additional low-pass filter has been applied to the data of a) to smooth the evolution and to reduce the impact of outliers. For the sake of clearness, the plots show only paths with a minimum length of 50 blocks.

merging, paths with missing blocks are part of the detected path set. Here, a cubic spline interpolation was applied to estimate the values of these missing blocks.

Most of the Doppler shifts increase in value over time by becoming more negative.. This is caused by the increasing speed (and therefore higher absolute Doppler shift) of the aircraft during take-off and by the relative movement of the aircraft with respect to the reflectors like airport buildings, trees and fences.

The figure shows no paths with a positive Doppler shift. This matches our observations from Fig. 3, where no strong MPCs with a positive Doppler shift were detected. In both cases, the underlying data was recorded during the same take-off; however, the data shown in Fig. 3 was recorded shortly after the data presented in Fig. 5.

The Doppler shifts, especially the shifts of the paths with the lowest Doppler shifts, show a step-wise unsteady evolution. From the plot this step size can be determined as 25.4 Hz which corresponds to the Doppler resolution of the DDSFs the underlying MPCs got extracted from. We therefore consider these steps not as a natural effect, but a processing artifact that will be addressed below.

The amplitudes of the detected components, including the LOS component given in orange, show a typical small scale fading behavior. We assume this fading of the individual components is caused by non-resolvable MPCs that add up either constructively or destructively.

2) Filtered Data Points: Fig. 5(b) shows low-pass filtered versions of the paths presented in Fig. 5(a). The purpose of the filtering is to smooth the curves to compensate the effects introduced by the processing, e.g. mis-detection of MPCs in the DDSF or, most prominent, the unsteady evolution of the Doppler shifts due to the low resolution. Another cause could be erroneously assigned MPCs: If d_{thresh} is chosen very large, some MPCs may be assigned to the same path although they are of different origin and therefore should be assigned to separate paths.

As neither the aircraft nor the reflectors are expected to do sudden movements that would explain these effects, it is a reasonable assumption that these effects are just processing artifacts in most of the cases.

The impact of the filtering can be clearly seen in the plot of the Doppler shifts: The step-wise leaps that dominated the plotted paths in the unfiltered case are now gone and the curves look much smoother.

The effect on the delay and the amplitudes, respectively, is not as strong as for the Doppler shifts; however, the paths now look smoother than before.

C. Reflector Localization

To validate the processing described above, the results of the MPC processing and tracking have been used to perform

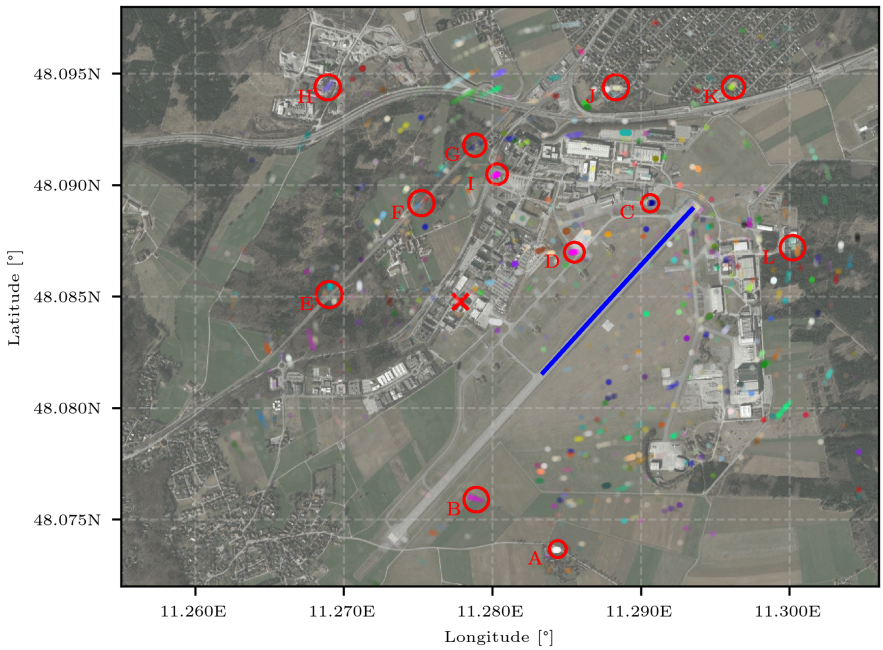

Fig. 6. Map of the EDMO airport with estimated positions of reflectors: The red cross marks the position of the transmitter, the blue line on the runway shows the track of the aircraft during the recording of the displayed data. Each reflectors position was estimated by evaluating the nodes of the corresponding path as given in Fig. 5b) according to Appendix E. Only paths of a minimum length of 50 blocks were considered. The locations of a subset of the mostprominent reflectors are highlighted by red circles. Background image: © by Google.

a reflector localization according to Appendix E. The estimated reflectors' locations are then compared to the actual location of potential reflectors given in a map.

Fig. 6 shows a map of the EDMO airport. The position of the transmitter of the channel sounding signal is marked by a red cross. The track the aircraft has traveled while the data that got evaluated for the reflector localization has been recorded is denoted by a blue line; the aircraft started heading south-west.

The path based reflector localization works by superimposing the results of the reflector localization for each individual MPC of an MPC path. The longer a path is, i.e. the more persistent an MPC is, the better will be the result of the estimation, as outliers and even ambiguities are averaged out due to the superimposition. In the figure, the certainty about the position of a reflector is represented by the data point's opacity. A color code is used to let the reader distinguish, what data points belong to a common path. However, as the amount of detected paths exceeds the amount of available colors by far, the colors are not exclusively mapped to a specific path.

While the comparatively high delay resolution allows a pretty precise estimation of the ellipse the reflector is located on, the estimation of the angle suffers from the comparatively low Doppler resolution. Although the superimposition helps to improve the precision of the angle estimation, the shape of the data point clusters shows the lack of precision of the angle estimation. However, in most of the cases the algorithm was able to resolve the ambiguities caused by the Doppler-to-angle conversion: The actual reflector's position has been identified, while the mirrored reflector's position got averaged out.

In the following, we want to discuss a few of the most prominent reflectors highlighted in the map.

The reflector at *A* in the south has been identified as a shelter with a metal roof. The reflectors around *B* have been identified as a fence that is part of the airfield barrier.

Many reflectors have also been found close to the aircraft along the runway: The reflectors both in *C* and *D* are very persistent and have been identified as airport service buildings.

The reflectors in *E*, *F*, and *G* are most likely related to the railway lines along which they have been detected. Their origin might be the power lines along the railway, the rails or even a train that passed by during recording.

A few reflectors have been detected in the mining area north of the highway at *H*; the reflector in *I* is located between a parking lot and an intersection with traffic lights.

The reflectors in *J* and *K* are located in the suburbs north of the highway and are most likely some higher buildings.

East of the runway we have highlighted the building in *L*, as it is also the origin of a quite persistent reflector.

VIII. Conclusion and Outlook

In this paper we have described a processing chain for the evaluation of channel sounding data recorded during flight. We have explained our pre-processing and the actual MPC extraction process before introducing a path-based MPC tracking approach. In a last step, we have applied our processing chain to our measurement data and performed a reflector localization based on the detected and tracked MPCs. We have shown that our approach can be used to locate reflectors that are causing MPCs that have a significant impact on the wireless channel.

In a next step we want to evaluate more of our measurement data recorded during other flight scenarios and use the results for a reflector localization – depending on the scenario also in three rather than in just two dimensions – and finally for channel modeling.

The authors are also planning to apply the presented MPC tracking approach to measurement data collected during other campaigns, not necessarily limited to the aeronautical air-ground channel.

APPENDIX A

A. Calculation of the Impulse Response

The Impulse Response (IR) is computed by correlating the transmitted signal, here denoted by the reference signal $\boldsymbol{x}_{\text{ref}} \in \mathbb{C}^N$, and one instance of the channel sounding sequence in the measurement data, either denoted by $\boldsymbol{y} \in \mathbb{C}^{BN}$ or $\boldsymbol{Y} \in \mathbb{C}^{N\times B}$. We implement the correlation in frequency domain, and assume a block processing of B consecutive impulse responses.[2] The computation of the IR either maps $\mathbb{C}^{BN} \mapsto \mathbb{C}^{Nf_{\text{up}}\times B}$ or $\mathbb{C}^{N\times B} \mapsto \mathbb{C}^{Nf_{\text{up}}\times B}$, depending on the input data. In case of a vector input, $\boldsymbol{y} \in \mathbb{C}^{BN}$ is converted to a matrix $\boldsymbol{Y} \in \mathbb{C}^{N\times B}$ where the item in the n-th row and the m-th column $y_{n,m}$ is set according to $y_{n,m} = \boldsymbol{y}[mB + n]$.

After this optional reshaping, the processing continues for both cases as follows.

1) A matrix $\boldsymbol{X}_{\text{ref}} \in \mathbb{C}^{N\times B}$ which elements are set to $x_{n,m} = \boldsymbol{x}_{\text{ref}}[n]\ \forall\ m \in \{0, 1, \ldots, B-1\}$ is created based on the reference signal; the Fast Fourier Transform (FFT) is computed: $\boldsymbol{X}'_{\text{ref}} := \mathfrak{FF}\mathfrak{T}_{N,\downarrow}\{\boldsymbol{X}_{\text{ref}}\}$.
2) The FFT along each column of $\boldsymbol{Y}$ is computed, resulting in a matrix $\boldsymbol{Y}' := \mathfrak{FF}\mathfrak{T}_{N,\downarrow}\{\boldsymbol{Y}\}$.
3) An element-wise multiplication $\boldsymbol{K}' := \boldsymbol{Y}' \cdot \boldsymbol{X}'_{\text{ref}}$ is performed; the result is expanded by $N(f_{\text{up}} - 1)$ rows of zeros in case upsampling is requested.
4) We then perform an Inverse Fast Fourier Transform (IFFT): $\boldsymbol{K} := \mathfrak{IFF}\mathfrak{T}_{N,\downarrow}\{\boldsymbol{K}'\}$.

The n-th column of $\boldsymbol{K} \in \mathbb{C}^{Nf_{\text{up}}\times B}$ now corresponds to the IR $\boldsymbol{h}_n$ of the n-th instance of the channel sounding sequence in the input data, upsampled by f_{up}.

Assuming that $\boldsymbol{x}_{\text{ref}}$ and $\boldsymbol{y}$ or $\boldsymbol{Y}$, respectively, are sampled at a rate of f_{sr}, the delay resolution of the IRs is given by

$$\Delta_\tau := (f_{\text{sr}} f_{\text{up}})^{-1}. \tag{22}$$

B Calculation of the Power Delay Profile

We denote the function computing the Power Delay Profile (PDP) by $\text{PDP}^{f_{\text{up}}}_{\boldsymbol{x}_{\text{ref}}}$. It either maps $\mathbb{C}^{BN} \mapsto \mathbb{C}^{Nf_{\text{up}}}$ or $\mathbb{C}^{N\times B} \mapsto \mathbb{C}^{Nf_{\text{up}}}$, where $\boldsymbol{x}_{\text{ref}}$ denotes the reference signal and f_{up} denotes the upsampling factor, depending on the input data.

The computation of the PDP is based on B consecutive Impulse Responses (IRs), thus we assume the IR matrix $\boldsymbol{K}$ has been computed according to Appendix A.

[2]The computation of a single IR therefore corresponds to the special case of $B = 1$.

The complex-valued PDP is then computed by taking the mean of matrix $\boldsymbol{K}$ along its rows: $\boldsymbol{a}_{\mathbb{C}} := \bar{\boldsymbol{K}}_{\rightarrow}$.

The real-valued PDP is given by the complex-valued PDP's absolute value: $\boldsymbol{a}_{\mathbb{R}} := |\bar{\boldsymbol{K}}_{\rightarrow}|$.

Please note that the definitions above correspond to the coherent PDP. Its delay resolution is given by (22).

C Calculation of the Doppler-Delay Spreading Function

We denote the discrete Doppler-Delay Spreading Function (DDSF), sometimes also called Cross Ambiguity Function (CAF), using reference signal $\boldsymbol{x}_{\text{ref}}$ and upsampling f_{up} by $\mathcal{S}^{f_{\text{up}}}_{\boldsymbol{x}_{\text{ref}}}$. It either maps $\mathbb{C}^{BN} \mapsto \mathbb{C}^{Nf_{\text{up}}\times B}$ or $\mathbb{C}^{N\times B} \mapsto \mathbb{C}^{Nf_{\text{up}}\times B}$, depending on the input data.

First, the B Impulse Responses (IRs) are computed according to Appendix A, resulting in the IR matrix $\boldsymbol{K}$.

Then, an FFT along the rows of $\boldsymbol{K}$ is performed: $\boldsymbol{A} := \mathfrak{FF}\mathfrak{T}_{B,\rightarrow}\{\boldsymbol{K}\}$. We assume that this FFT includes an FFT-shift, such that the 0-th frequency is shifted to column $B/2$.

The columns of the resulting matrix $\boldsymbol{A}$ correspond to the delay axes, the rows correspond to the Doppler axes. While the resolution along the delay axis is the same as for the IRs as given in (22), the resolution along the Doppler axis is given by

$$\Delta_\nu := \frac{f_{\text{sr}}}{NB}. \tag{23}$$

D. Distance Metrics

The function $\mathfrak{D} : \mathbb{R}^4 \times \mathbb{R}^4 \mapsto \mathbb{R}$ returns the distance metric between two Multipath Components (MPCs) ξ_n and ξ_m.

The first step is to perform a feature scaling, as the typical values and units of the items of an MPC vary. If not denoted otherwise, we apply the popular min-max scaling to all elements x of an MPC:

$$\forall\ x \in \xi :$$
$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \tag{24}$$

where $x_{\{\min,\max\}}$ denote a minimum/maximum value, respectively, that are used for the scaling of the specific item. They are chosen such that all appearing values of the specific item are mapped to a value within $[0, 1)$.

1) Delay-Only Metric: In this case, only the MPCs' delay is considered. This metric is useful in case no Doppler information is available and huge variations in the MPCs' amplitude are expected. We define it as

$$\mathfrak{D}\{\xi_n, \xi_m\} = \sqrt{(\tau'_n - \tau'_m)^2}, \tag{25}$$

where $\tau'_{\{n,m\}}$ are feature scaled (24) versions of $\tau_{\{n,m\}} \in \xi_{\{n,m\}}$.

2) Delay-Doppler-Only Metric: In this case, the MPCs' complex weight is ignored. This metric is useful in case huge variations in the MPCs' amplitude are expected. We define it as

$$\mathfrak{D}\{\xi_n, \xi_m\} = \sqrt{(\tau'_n - \tau'_m)^2 + (\nu'_n - \nu'_m)^2}, \tag{26}$$

where $\{\tau_{\{n,m\}}{}', \nu'_{\{n,m\}}\}$ are feature scaled (24) versions of $\{\tau_{\{n,m\}}, \nu_{\{n,m\}}\} \in \xi_{\{n,m\}}$.

E. Two-Dimensional Reflector Localization

The delay and Doppler information of an MPC can be used to estimate the corresponding reflector's location. Here, we briefly describe the implementation of the reflector localization for the two-dimensional case (i.e. no height information is provided).

1) Exploiting Delay Information: All (potential) reflectors with the same delay are located on an ellipse whose focal points are given by the position of the transmitter (Tx) and the receiver (Rx). It is straight forward to compute this set of points based on a given delay τ: $\mathcal{E}_\tau = \{P \mid \overline{\mathrm{Tx}P} + \overline{P\mathrm{Rx}} = c_{\mathrm{air}}\tau\}$,

$$\mathcal{E}_\tau = \left\{P \mid \overline{\mathrm{Tx}P} + \overline{P\mathrm{Rx}} = c_{\mathrm{air}}\tau\right\},$$

where c_{air} denotes the speed of light in air.

If we assume an MPC-detection with a delay resolution of Δ_τ, we can expand this to an inner and an outer ellipse. The reflector's position is then lying in the set:

$$\mathcal{E}_\tau = \left\{P \;\middle|\; \overline{\mathrm{Tx}P} + \overline{P\mathrm{Rx}} \lessgtr c_{\mathrm{air}}\left(\tau \pm \frac{\Delta_\tau}{2}\right)\right\}. \tag{27}$$

2) Exploiting Doppler Information: The angle of arrival α with respect to the receiver's heading and speed $\vec{v}_{\mathrm{Rx}}$ can be estimated based on the Doppler shift ν according to:

$$\alpha = \pm \arccos\left\{\frac{c_{\mathrm{air}}\nu}{f_{\mathrm{c}}|\vec{v}_{\mathrm{Rx}}|}\right\}. \tag{28}$$

The reader may note the non-resolvable ambiguity caused by the nature of the cosine function. In case of a Doppler resolution of Δ_ν, the set of points that arrive at the receiver within the given angle range is given by

$\mathcal{E}_\nu = \mathcal{E}_{-\nu} \cup \mathcal{E}_{+\nu}$, where

$$\mathcal{E}_{-\nu} = \left\{P \;\middle|\; \angle(\overline{P\mathrm{Rx}}, \vec{v}_{\mathrm{Rx}}) \lessgtr -\arccos\left\{\frac{c_{\mathrm{air}}(\nu \mp \frac{\Delta_\nu}{2})}{f_{\mathrm{c}}|\vec{v}_{\mathrm{Rx}}|}\right\}\right\}$$

and

$$\mathcal{E}_{+\nu} = \left\{P \;\middle|\; \angle(\overline{P\mathrm{Rx}}, \vec{v}_{\mathrm{Rx}}) \lessgtr +\arccos\left\{\frac{c_{\mathrm{air}}(\nu \pm \frac{\Delta_\nu}{2})}{f_{\mathrm{c}}|\vec{v}_{\mathrm{Rx}}|}\right\}\right\}. \tag{29}$$

3) Localization: The final position of the reflector is lying in the intersection of the sets $\mathcal{E} = \mathcal{E}_\tau \cap \mathcal{E}_\nu$. By combining the information gained through the processing of multiple blocks recorded over time, the uncertainty over the reflector's position can be reduced.

4) Exploiting Evolution: As it is assumed that an MPC's signal is caused by a reflector, its localization can be improved by observing the evolution of the corresponding MPC represented by a path γ. This can be done by performing the steps described above for each node of the path and finally joining these results: $\mathcal{E}_\gamma = \bigcap_{\forall b \in \gamma} \mathcal{E}^{(b)}$.

5) Implementation: In practice, the sets that are used to locate the reflector are represented by a two-dimensional matrix, where the rows correspond to the east-west axis and the columns correspond to the north-south axis of an East-North-Up (ENU) coordinate system with the transmitter in its origin. For each node of an MPC's path, the matrix is determined as given above. Instead of computing the intersection of the sets as given in the last step, the exploitation of the evolution to reduce ambiguities is realized by computing the mean of the matrices of each path node. This approach is more robust against outliers as erroneous detections are averaged out.

REFERENCES

[1] Fortune Business Insights, "Unmanned aerial vehicle (UAV) market size, share & COVID-19 impact analysis," Jul. 2020. [Online]. Available: https://www.fortunebusinessinsights.com/industry-reports/unmanned-aerial-vehicle-uav-market-101603

[2] M. Webster, M. Fisher, N. Cameron, and M. Jump, "Formal methods for the certification of autonomous unmanned aircraft systems," in *Proc. IEEE 30th Int. Conf.*, 2011, pp. 228–242.

[3] "Characteristics of unmanned aircraft systems and spectrum requirements to support their safe operation in non-segregated airspace," Int. Telecommun. Union, Geneva, Switzerland, Tech. Rep. ITU-R M.2171, 2009.

[4] M. Pätzold, *Mobile Fading Channels*. New York, NY, USA: Wiley, 2002.

[5] D. W. Matolak and R. Sun, "Air-ground channel characterization for unmanned aircraft systems - Part I: Methods, measurements, and models for over-water settings," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 26–44, Jan. 2017.

[6] R. Sun and D. W. Matolak, "Air-ground channel characterization for unmanned aircraft systems - Part II: Hilly and mountainous settings," *IEEE Trans. Veh. Technol.*, vol. 66, no. 3, pp. 1913–1925, Mar. 2017.

[7] D. W. Matolak, "Air-ground channels & models: Comprehensive review and considerations for unmanned aircraft systems," in *Proc. IEEE Aerosp. Conf.*, 2012, pp. 1–17.

[8] P. Hoeher and E. Haas, "Aeronautical channel modeling at vhf-band," in *Proc. Gateway to 21st Century Commun. Village. VTC 1999-Fall. IEEE VTS 50th Veh. Technol. Conf.*, vol. 4, 1999, pp. 1961–1966.

[9] E. Haas, "Aeronautical channel modeling," *IEEE Trans. Veh. Technol.*, vol. 51, no. 2, pp. 254–264, Mar. 2002.

[10] P. Pulini, "Forward link performance analysis for the future ieee 802.16-based airport data link," in *Proc. IEEE Int. Conf. Commun.*, 2010, pp. 1–5.

[11] I. Sen and D. W. Matolak, "The 5-Ghz airport surface area channel - Part II: Measurement and modeling results for small airports," *IEEE Trans. Veh. Technol.*, vol. 57, no. 4, pp. 2027–2035, Jul. 2008.

[12] N. Schneckenburger, T. Jost, D. Shutin, M. Walter, G. del Galdo, and U. Fiebig, "Reflector localization for geometrical modeling the airground channel," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 7994–8008, Sep. 2018.

[13] N. Schneckenburger, T. Jost, M. Walter, G. del Galdo, D. W. Matolak, and U. Fiebig, "Wideband airground channel model for a regional airport environment," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6243–6256, Jul. 2019.

[14] J. Kim and I. Lee, "Channel measurements and characterizations for long range air-to-ground communication systems in the UHF band," *IEEE Access*, vol. 10, pp. 101880–101888, 2022.

[15] D. W. Matolak and R. Sun, "Air–ground channel characterization for unmanned aircraft systems–Part III: The suburban and near-urban environments," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 6607–6618, 2017.

[16] D. M. Mielke, N. Schneckenburger, U.-C. Fiebig, M. Walter, and M. A. Bellido-Manganell, "Analysis of the dominant signal component of the air-ground channel based on measurement data at C-Band," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 2955–2968, Oct. 2021.

[17] N. Schneckenburger, "A wide-band air-ground channel model," Ph.D. dissertation, Fakultät für Elektrotechnik und Informationstechnik, Univ. Ilmenau, Ilmenau, Germany, 2017.

[18] M. Powell, "The BOBYQA algorithm for bound constrained optimization without derivatives," Tech. Rep. DAMTP 2009/NA06, 2009. [Online]. Available: https://www.damtp.cam.ac.uk/user/na/NA_papers/NA2009_06.pdf

[19] D. Shutin and G. Kubin, "Tracking and prediction of multipath components in wireless MIMO channels," in *Proc. 65th IEEE Veh. Technol. Conf.*, 2007, pp. 229–333. [Online]. Available: papers/vtc2007.pdf

[20] M. Walter, D. Shutin, M. Schmidhammer, D. W. Matolak, and A. Zajic, "Geometric analysis of the doppler frequency for general non-stationary 3D mobile-to-mobile channels based on prolate spheroidal coordinates," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 10419–10434, Oct. 2020.

Daniel M. Mielke received the B.Sc. and the M.Sc. degrees from the University of Kiel, Kiel, Germany, in 2014 and 2016, respectively, and the Ph.D. degree from the University of Ulm, Ulm, Germany. Since 2016, has been with the Institute of Communications and Navigation, German Aerospace Center (DLR), Wessling, Germany His research interests include radio wave propagation and channel modeling in the field of aeronautical communications, signal processing, and robust communication systems.

Michael Walter (Senior Member, IEEE) received the Dipl.-Ing. and Dr.-Ing. degrees in electrical engineering from Ulm University, Ulm, Germany, in 2008 and 2015, respectively. Since 2009, he has been the Member of the Scientific Staff with the Institute of Communications and Navigation, German Aerospace Center (DLR),, Wessling, Germany, where he gained extensive experience in organizing and conducting numerous measurement campaigns for vehicular and aeronautical research.

Dennis Becker received the bachelor's degree in electrical engineering from Duale Hochschule Baden-Württemberg, Stuttgart, Germany, in 2011, the master's degree from Technische Universität Darmstadt, Darmstadt, Germany, in 2014, and the master's degree in information and communications engineering in 2017. In January 2018, Dennis joined the Institute of Communication and Navigation at the German Aerospace Center (DLR), Wessling, Germany as a Research Assistant with the Aeronautical Communications Group. His research focuses on analyzing signal propagation characteristics of the communications channel between small sized unmanned aerial vehicles.

Miguel A. Bellido-Manganell received the B.Sc. degree in telecommunication engineering from University of Granada, Granada, Spain, in 2014, and the M.Sc. degree in telecommunication engineering from the University of Malaga, Malaga, Spain, in 2016. In 2016, he joined the Institute of Communications and Navigation of the German Aerospace Center (DLR), Wessling, Germany, as a Research Associate. He has participated in multiple measurement campaigns and had a leading role in the 2019 LDACS flight campaign. His research interests include channel modeling, physical layer design and digital signal processing in wireless communications, and medium access control in ad hoc networks.

Uwe-Carsten Fiebig (Senior Member, IEEE) received the Electrical Engineering degree from the Technical University of Munich, Munich, Germany, and the Dr.-Ing. degree from the University of Kaiserslautern, Kaiserslautern, Germany, in 1993. He spent the last year of his studies with ESIEE, Paris, France. . He joined the Institute of Communications and Navigation with German Aerospace Research (DLR), Wessling, Germany, in 1988. As the Head of Department, he is responsible for research in aeronautical and vehicular communications, multi-sensor navigation and swarm exploration. A special focus is on propagation and communications for aircraft, drones, ships and trains. Since 2011, he has been an Honorary Professor with the University of Ulm, Ulm, Germany, and teaches satellite communication and navigation.