Learning-Based Fusion of Multipath Assisted Positioning and Fingerprinting

Markus Ulmschneider, Christian Gentner and Armin Dammann
German Aerospace Center (DLR), Institute of Communications and Navigation

BIOGRAPHIES

Markus Ulmschneider received his Master's degree in 2014 from the University of Ulm, and his PhD (Dr.-Ing.) from the Hamburg University of Technology in 2021. Since 2014, he has been working at the Institute of Communications and Navigation of the German Aerospace Center (DLR), focusing on multipath assisted positioning.

Christian Gentner received his M.Sc. and Dr.-Ing. (Ph.D.) degrees in electrical Engineering in 2009 and 2018, respectively, from the University of Ulm. Since 2009, he is working at the Institute of Communications and Navigation of the German Aerospace Center (DLR). His current research focuses on multipath assisted and indoor positioning.

Armin Dammann received his Diploma (Dipl.-Ing.) and PhD (Dr.-Ing.) from the University of Ulm in 1997 and 2005 respectively. In 1997 he joined the Institute of Communications and Navigation of the German Aerospace Center (DLR). Since 2005 he is head of the Mobile Radio Transmission research group of the institute.

ABSTRACT

In multipath assisted positioning, multipath components (MPCs) are regarded as line-of-sight (LoS) signals from virtual transmitters. The locations of the physical and the virtual transmitters can be estimated jointly with the user position using simultaneous localization and mapping (SLAM). We have previously introduced such an approach called cooperative Channel-SLAM, where multiple users cooperatively estimate the locations of physical and virtual transmitters. Such schemes typically suffer from a high computational complexity due to expensive signal processing, though. Within this paper, we propose a novel approach that combines multipath assisted positioning with fingerprinting. In the first stage, multiple users estimate their own locations with cooperative Channel-SLAM. With the channel estimates and the estimated user positions from cooperative Channel-SLAM, a deep neural network (DNN) is trained. In the second stage, users can localize themselves making use of the DNN. In our novel approach, the positioning error is in the same order of magnitude as for cooperative Channel-SLAM, while the computational complexity is reduced drastically.

I. INTRODUCTION

In multipath assisted positioning schemes, multipath propagation is exploited for localizing a user. Scattered and reflected signal components are regarded as line-of-sight (LoS) signals from so called virtual transmitters. While the locations of these virtual transmitters depend on the scenario and are typically unknown, their location can be estimated jointly with the user location with simultaneous localization and mapping (SLAM) [1–3]. We have recently introduced a multipath assisted positioning approach named cooperative Channel-SLAM in [4], where users cooperate by exchanging maps of estimated transmitter positions. Single user Channel-SLAM works in two steps. In the first step, a channel estimator estimates and tracks the parameters of signal components of the received signal over time. In the second step, these channel parameters are used in a Rao-Blackwellized particle filter to jointly track the user location and estimate the positions of the physical and the virtual transmitters. In cooperative Channel-SLAM, multiple users going through the same scenario cooperate by exchanging maps of estimated transmitter positions. Each user is in a local coordinate system, though, which has an unknown relative rotation and translation to the coordinate systems of other users. Estimating these transformation parameters is a crucial element of cooperative Channel-SLAM and denoted by the term map matching.

We have shown before that the positioning error [4] and the entropy of the maps [5] converge in cooperative Channel-SLAM. Nevertheless, the scheme suffers from a high complexity both in the channel estimation, the particle filter and map matching. For channel estimation in cooperative Channel-SLAM, we use the Kalman Enhanced Super Resolution Tracking (KEST) [6], which tracks the parameters of signal components over time with parallel Kalman filters. Each Kalman filter corresponds to a hypothesis on the model order. In this way, KEST establishes correspondences among signal components for successive time instants, yielding inherently a short time data association. Though, KEST takes hard decisions regarding the model order and this short-term data association. False associations can therefore considerably deteriorate the positioning performance.

A different approach to indoor localization are fingerprinting schemes [7, 8], which work in two stages. In the first stage called offline phase, features at known locations are collected and stored in a database. Such features could be information on the wireless channel or magnetic signatures, for example, and they are called fingerprints. When using wireless radio signals such as wireless local area network (WLAN) or Bluetooth, received signal strength indicator (RSSI) values or channel state information (CSI) are often used as fingerprints. In the second stage, the online phase, a user can be localized by matching fingerprints taken at their location against the database.

Fingerprinting methods with wireless radio signals can easily be build on top of existing infrastructure, such as WLAN systems. The key assumption is that certain features of radio signals correlate with certain locations. Fingerprinting is hence a data-driven approach and does not require complex models.

A major drawback of fingerprinting schemes is the tedious collection of features in the offline phase. For good localization performance, fingerprints need to be collected at many positions. Even more, a precise location needs to be stored for every fingerprint, requiring a precise localization ability in the offline phase. In addition, changes in the environment can decrease the positioning performance significantly. Updating the fingerprint database to adapt to such changes may take a lot of effort. Some research therefore aims at updating fingerprints in the online phase, e.g. [9].

Recently, techniques from machine learning have been used to improve the performance and compensate for the disadvantages of fingerprinting, for example k-nearest neighbours (kNN) [10], support vector machine (SVM) and deep neural networks (DNNs) [11]. In the offline phase of the latter example, fingerprints and the user location are in principle used to train a DNN. In the online phase of fingerprinting, evaluation of the neural network yields the user location. Compared to competing approaches, DNNs tend to have better generalization abilities [12].

Within this paper, we propose a novel hybrid scheme combining cooperative Channel-SLAM with fingerprinting methods using a DNN. The scheme works in two stages and will be donated by the term DNN-CC-SLAM in the following. In the first stage, users navigate through a scenario with cooperative Channel-SLAM. The time of arrival (ToA) estimates from the channel estimators together with the respective estimated positions are stored in a fingerprint database. After a sufficient number of ToA fingerprints are available, a DNN is trained to predict the user location from these estimated ToAs. Hence, the fingerprints are not collected in a classical way with a ground truth positioning system, but are nuisance parameters from cooperative Channel-SLAM. In particular, no external hardware or equipment needs to be installed for collecting or updating the fingerprints. The fingerprint collection can be regarded as a crowd-sourcing scheme.

In the second stage, further users can exploit the trained DNN for positioning. For estimating the ToAs of signal components in the online phase, estimators with low complexity exist. Hence, the second stage of DNN-CC-SLAM is crucially less complex than cooperative Channel-SLAM.

Compared to standard fingerprinting approaches, we facilitate the tedious collection of features in the fingerprinting scheme with external infrastructure for a precise localization reference. The fingerprint database can be updated with cooperative Channel-SLAM from time to time or when needed, for example by one or more dedicated users or by a certain percentage of users navigating in the scenario.

II. COOPERATIVE CHANNEL-SLAM

Fig. 1 illustrates the idea behind multipath assisted positioning with one physical transmitter labeled Tx. For clarity, a possible LoS propagation path is omitted. Reflected and scattered multipath components (MPCs) are regarded as LoS signals from the virtual transmitters vTx1 and vTx2. If the signal is reflected at a straight surface such as the wall, the respective virtual transmitter vTx1 is at the location of the physical transmitter mirrored at the wall. Likewise, if the signal is scattered at the point scatterer, it is regarded as a LoS signal from vTx2, which is located at the scatterer's location.

Geometrical considerations reveal three important properties of virtual transmitters. First, in the case of reflections of the signal at straight surfaces and scattering at point scatterers, the location of the corresponding virtual transmitters is independent from the user position. Second, in the case of reflection at straight walls, the respective virtual transmitter is time synchronized to the physical transmitter. In the case of scattering, there is a delay offset between the physical and the virtual transmitter corresponding to the actual propagation time of the signal from the physical to the virtual transmitter. This delay offset can be interpreted as a clock offset. Third, the concept of single interactions of the signals with straight surfaces and point scatterers can be extended to multiple interactions in a straightforward manner [13].

Channel-SLAM assumes a linear multipath channel with static physical transmitters and a mobile user. In the following, we will assume only one physical transmitter for clarity, while the generalization to multiple physical transmitters is straightforward if the signals of the physical transmitters can be separated in at least one domain such as time or frequency. The movement of the user equipped with the receiver makes the channel time variant. The channel estimator KEST works in two stages. In the inner stage, a maximum likelihood (ML) parameter estimator such as Space-Alternating Generalized Expectation-Maximization (SAGE) estimates the parameters of signal components such as ToA, phase, amplitude, and angle of arrival (AoA). In the outer

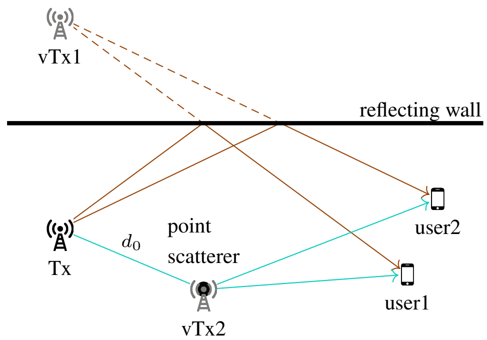

Figure 1: The fundamental idea in multipath assisted positioning is to treat every MPC as a LoS signal from a virtual transmitter.

stage, KEST uses parallel Kalman filters to track these parameters over time. Each Kalman filter corresponds to one hypothesis on the model order, i.e., the number of signal components. Thus, KEST tracks both the number and the the parameters of signal components over time, and establishes correspondences among signal components for neighboring time instants.

We assume that the channel is constant for the length of one snapshot. A snapshot of the received signal is sampled at time instants k at the receiver and fed into KEST. The resulting parameter estimates, typically ToA and AoA, are stored in the vector

$$\boldsymbol{z}_k = \left[\mathrm{ToA}_{1,k}\ \mathrm{AoA}_{1,k}\ \ldots\ \mathrm{ToA}_{N_{\mathrm{TX},k},k}\ \mathrm{AoA}_{N_{\mathrm{TX},k},k}\right], \tag{1}$$

where $\mathrm{ToA}_{j,k}$ and $\mathrm{AoA}_{j,k}$ are the ToA and AoA of the j^{th} signal component, and $N_{\mathrm{TX},k}$ is the number of signal components detected by KEST at time instant k. Since each signal component corresponds to one transmitter in Channel-SLAM, $N_{\mathrm{TX},k}$ is also the number of transmitters at time instant k. Note that Channel-SLAM does not differentiate between physical and virtual transmitters. In principle, Channel-SLAM works with ToA or AoA estimates only in $\boldsymbol{z}_k$. However, combining ToA and AoA information significantly reduces the convergence time and the computational complexity, and improves the positioning performance.

The state $\boldsymbol{x}_{\mathrm{u},k}$ of the user at time instant k consists of their two-dimensional position $\boldsymbol{p}_{\mathrm{u},k}^T$ and velocity $\boldsymbol{v}_{\mathrm{u},k}^T$. Since the transmitters are assumed static, the state $\boldsymbol{x}_{\mathrm{TX},k}^{<j>}$ of the j^{th} transmitter at time instant k consists of its location $\boldsymbol{p}_{\mathrm{TX},k}^{<j>}$ and its clock offset $\tau_{0,k}^{<j>}$. The joint state of all $N_{\mathrm{TX},k}$ transmitters at time instant k is denoted by $\boldsymbol{x}_{\mathrm{TX},k}$.

Channel-SLAM jointly estimates the history of the state of the user and of the states of the transmitters based on the estimates from the channel estimator and possibly control input, which may integrate additional sensors such as a gyroscope or an inertial measurement unit (IMU). The joint user and transmitter state at time instant k is denoted by $\boldsymbol{x}_k$. The respective state probability density function (PDF) for the history of this joint state can be expressed as

$$\begin{aligned} \mathrm{p}\left(\boldsymbol{x}_{0:k}|\boldsymbol{z}_{1:k},\boldsymbol{u}_{1:k}\right) &= \mathrm{p}\left(\boldsymbol{x}_{\mathrm{TX},0:k},\boldsymbol{x}_{\mathrm{u},0:k}|\boldsymbol{z}_{1:k},\boldsymbol{u}_{1:k}\right) \\ &= \mathrm{p}\left(\boldsymbol{x}_{\mathrm{u},0:k}|\boldsymbol{z}_{1:k},\boldsymbol{u}_{1:k}\right)\mathrm{p}\left(\boldsymbol{x}_{\mathrm{TX},0:k}|\boldsymbol{x}_{\mathrm{u},0:k},\boldsymbol{z}_{1:k}\right), \end{aligned} \tag{2}$$

where $\boldsymbol{x}_{0:k}$ describes the history of the variable $\boldsymbol{x}$ from time instants zero to k, and $\boldsymbol{u}_{1:k}$ is the control input.

The last factor in the last line of Eq. (2) is the PDF regarding the transmitter states. We assume that the estimates from the channel estimator for one signal component is uncorrelated with the estimates for other signal components, or that such a correlation is present for only a very short time. Hence, the transmitters are assumed independent from each other and the transmitter state PDF can be factorized as

$$\mathrm{p}\left(\boldsymbol{x}_{\mathrm{TX},0:k}|\boldsymbol{x}_{\mathrm{u},0:k},\boldsymbol{z}_{1:k}\right) = \prod_{j=1}^{N_{\mathrm{TX},k}} \mathrm{p}\left(\boldsymbol{x}_{\mathrm{TX},0:k}^{<j>}|\boldsymbol{x}_{\mathrm{u},0:k},\boldsymbol{z}_{1:k}\right). \tag{3}$$

Estimating the joint user and transmitter state is done in Channel-SLAM with a Rao-Blackwellized particle filter. The user state is approximated by a number of weighted samples in the user state space. For each user particle in the user particle filter, the state of each transmitter is estimated with a transmitter particle filter independently from the other transmitters following the factorization in Eq. (3).

The set of transmitter states estimated by a user is denoted by the term map in the following. When multiple users navigate in the same area with Channel-SLAM, they can share such maps to obtain prior knowledge on the transmitters' states. A map can

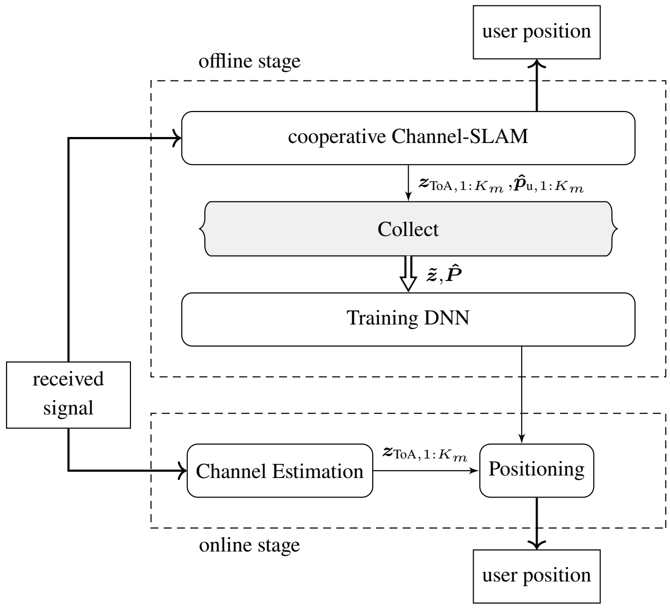

Figure 2: The flow chart of DNN-CC-SLAM.

be exploited by each user, and also improved by adding new transmitters to the map and improve transmitter state estimates with own measurements. In this cooperative Channel-SLAM scheme, the relative rotations and translations among the coordinate systems of the users need to be estimated, and correspondences among transmitters in different user maps need to be established. Estimating these transformation parameters and correspondences is denoted by the term map matching [14], and is an essential element of cooperative Channel-SLAM.

Map matching is performed by an adapted random sample consensus (RANSAC) algorithm based on estimated transmitter states in two maps. One challenge is that a transmitter in the one map may or may not have a correspondence to a transmitter in the other map, and vice versa. In a nutshell, the adapted RANSAC works iteratively in three steps. First, two sets of transmitters in both maps are chosen randomly. Second, the best set of correspondences among the transmitters is estimated with least squares (LS) based on the relative distances among the transmitters in each map. Third, based on these correspondences, the transformation parameters are estimated. These three steps are performed repeatedly until a sufficiently good map match was found or for a number of iterations.

In cooperative Channel-SLAM, map matching is performed not only once, but periodically as a user moves through a scenario. The more transmitters are in a map and the more converged their state estimates are, the better and more reliable becomes the map matching solution. Thus, an initial map matching solution is refined periodically. Once a map match is obtained, the user can localize themselves not only in their own local but in a common coordinate system, and both exploit and improve a shared map which is in this common system. Due to the rather high complexity of map matching and since there are no big changes in the transmitter state estimates expected from one time instant to another, map matching is performed not at every time instant.

III. FUSION OF COOPERATIVE CHANNEL-SLAM AND FINGERPRINTING

A flowchart of DNN-CC-SLAM is depicted in Fig. 2. In the first stage, users navigate through the scenario with cooperative Channel-SLAM described in Section II. Following the fingerprinting terminology, we may call this first stage the offline stage. From the channel estimators, the ToAs $\boldsymbol{z}_{\text{ToA},1:K_m}$ are obtained from each user as a subset of $\boldsymbol{z}_{1:K_m}$, where K_m is the number of time instants for the m^{th} user. Likewise, from the particle filter, we obtain the corresponding estimated positions of each user, namely $\hat{\boldsymbol{p}}_{\text{u},1:K_m}$. We collect and stack these estimated positions of all users at all time instants obtained with cooperative Channel-SLAM in the matrix $\hat{\boldsymbol{P}}$ such that each row of $\hat{\boldsymbol{P}}$ corresponds to one estimated position in two dimensions. We define the list $\tilde{\boldsymbol{z}}$ such that its s^{th} entry is denoted by $\tilde{\boldsymbol{z}}_s$ and contains the estimated ToAs for the estimated user position $\hat{\boldsymbol{P}}_s$, where $\hat{\boldsymbol{P}}_s$ is the s^{th} row of $\hat{\boldsymbol{P}}$. The values of $\tilde{\boldsymbol{z}}_s$ are obtained from the vectors $\boldsymbol{z}_{\text{ToA},1:K_m}$ for all users. These estimated ToAs in $\tilde{\boldsymbol{z}}_s$ for all s are considered the fingerprints. While the AoAs and further channel parameters may be used as well as fingerprints in principle, we do not use them as they would lead to a much more complex channel estimation in the online stage. In addition, using AoA

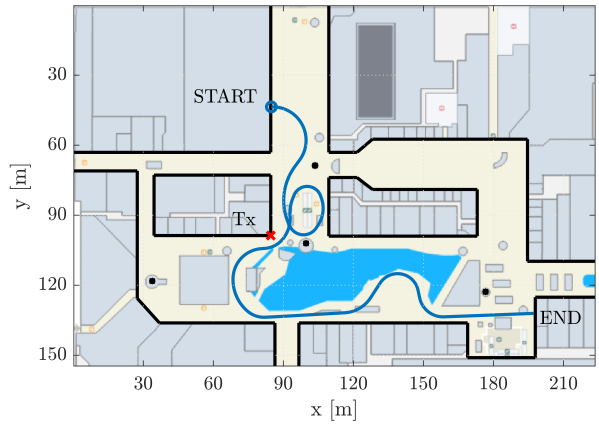

Figure 3: Top view of the indoor simulation scenario with the reference track. The only physical transmitter in the scenario is labeled Tx and depicted by the red cross.

for fingerprinting, complex orientation estimation issues would arise in the online stage.

Once the matrix $\hat{\boldsymbol{P}}$ has a certain size, i.e., once we have collected enough fingerprints, $\hat{\boldsymbol{P}}$ and $\tilde{\boldsymbol{z}}$ are used to train a DNN. Our DNN consists of an input layer, six hidden layers, and an output layer. All neighboring layers are fully connected. The number of neurons in the hidden layers are 1000, 300, 200, 100, 50 and 20. The input to the DNN are estimated ToAs in $\tilde{\boldsymbol{z}}$, the output is a position from $\hat{\boldsymbol{P}}$. Since the vectors $\tilde{\boldsymbol{z}}_s$ are in general of different lengths, zero padding is applied if needed. The activation function of all nodes in the DNN is the rectified linear unit (ReLU) activation function. The DNN is trained with the Adam optimizer [15] and the mean square error (MSE) as loss function. While the DNN can be trained as new training data from users arrive in a real implementation, we collect the training data before training the DNN first for clarity.

The reason to take only estimated ToAs into account instead of the entire received signal is the complexity in training the DNN. The idea is to extract relevant features of the received signal in order to keep the dimension of the DNN rather small. A very large input to the DNN would imply a more complex DNN and therefore an increased computational complexity in particular for training. Since a multipath channel typically is of sparse nature, we chose the ToAs of MPCs as relevant features. In particular, the ToAs are a nuisance parameter in cooperative Channel-SLAM and are available for training the DNN, keeping the additional complexity for the users low.

Once the DNN has been trained, further users in the scenario can use it for positioning in the second stage, or online stage, of DNN-CC-SLAM. While KEST is used as channel estimator in cooperative Channel-SLAM, i.e., during the offline phase of the fingerprinting scheme, we have found superfast line spectral estimation (SFLSE) [16] to be very useful for channel estimation in the online phase. SFLSE is a Bayesian method for line spectral estimation using sparse estimation techniques. It inherently estimates relevant model parameters such as the noise variances ands the model order. A big advantage of SFLSE is its very low complexity compared to other channel estimators, while keeping a comparable estimation accuracy [16]. The short-term data association provided by KEST is an important element for Channel-SLAM, but not necessary for fingerprinting. On our simulation machines, the SFLSE is on average about 75 times faster than KEST.

An inherent drawback of fingerprinting schemes and DNNs is their inability to extrapolate. When training a DNN, for example, the DNN's parameters are fitted to match the provided data. Consequently, the output of a DNN is unreliable in regions where no training data is available. Our hybrid scheme is based on crowd sourced data collection in cooperative Channel-SLAM. Accordingly, training data is collected in areas where many users tend to roam. In such areas, DNN-CC-SLAM is expected to perform much better than in areas where only few or even no users have collected training data.

IV. SIMULATIONS

To evaluate our novel approach, we performed simulations in an indoor scenario with a single physical transmitter. A top view of the scenario is depicted in Fig. 3. The physical transmitter is labeled Tx and depicted by the red cross. The black lines are walls representing planar surfaces, while the black dots are pillars representing point scatterers. In the scenario, a user track that we will refer to as the reference track is plotted by the blue line. The length of the reference track is $311.2\,\mathrm{m}$.

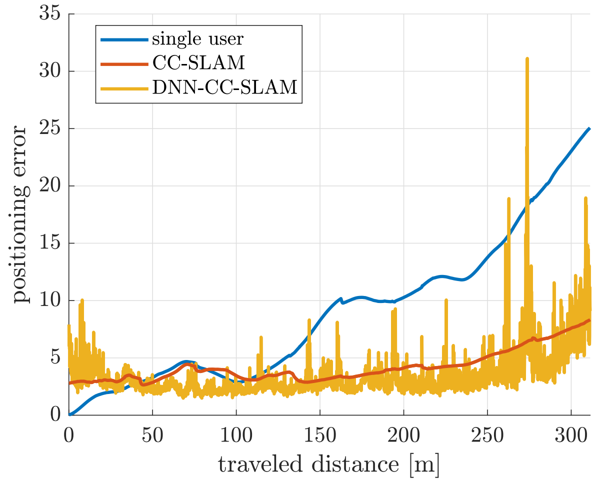

Figure 4: The positioning error of single user Channel-SLAM, cooperative Channel-SLAM and DNN-CC-SLAM.

The physical transmitter continuously transmits a signal known to the user. This transmit signal has a constant power spectral density over the entire bandwidth of $100\,\mathrm{MHz}$. Its carrier frequency is at $1.9\,\mathrm{GHz}$. A snapshot of the received signal is sampled every $100\,\mathrm{ms}$ at the receiver and processed by the channel estimator. The user is equipped with a rectangular antenna array of 9 elements arranged in a uniform 3×3 grid. AoA information is used in cooperative Channel-SLAM, but neither in training or evaluating the DNN. The angular resolution of the antenna array is in the order of a few degrees for well separated signal components. Similar to AoA information, turn rate information from a gyroscope that is rigidly mounted to the receiver is only used in the particle filter in cooperative Channel-SLAM as control input.

At each user position, the channel impulse response (CIR) is calculated with ray tracing to obtain a received signal for the simulations. However, the users do neither know the location of the physical transmitters nor have any information regarding the locations of walls and scatterers in the scenario. The initial position of the users is known in their own local coordinate system. However, the transformation parameters relating these local coordinate systems to those of other users are unknown and estimated with map matching in cooperative Channel-SLAM.

We have shown in [4] that the positioning error of cooperative Channel-SLAM converges in the order of 4-5 meters after 21 different users have gone through the scenario in Fig. 3. The trajectories of the 21 users sum up to a total traveled distance of $3.8\,\mathrm{km}$. To evaluate DNN-CC-SLAM, we simulated another $5.6\,\mathrm{km}$ of synthetic trajectories with cooperative Channel-SLAM corresponding to further 23 users. Overall, this leads to $8.9\,\mathrm{km}$ of trajectories of 44 different users as training data to feed the DNN.

The positioning performance a reference user going along the reference track in Fig. 3 is plotted in Fig. 4 in terms of the mean absolute error (MAE) for single user and cooperative Channel-SLAM and in terms of the Euclidean distance between the true and estimated positions for DNN-CC-SLAM. Due to the stochastic nature of the particle filter in Channel-SLAM and the optimization method in the DNN, all results in Fig. 4 are averaged over 100 runs.

The positioning performance of single user Channel-SLAM is depicted by the dark blue curve for reference. The positioning error along the reference track keeps increases over time. Only in the region around a traveled distance of $75\,\mathrm{m}$, we can make use of loop closure as the user is going along a loop, making the positioning error decrease to some extent. Note that in single user Channel-SLAM, no map matching is performed and the user is in their own local coordinate system. Accordingly, the positioning error is the error within that local reference system and is zero at the beginning of the track, making a comparison somewhat unfair.

The red curve depicts the positioning error for the cooperative Channel-SLAM case on the reference track, where 21 different users with a total traveled distance of $3.8\,\mathrm{km}$ have contributed to a prior map. The positioning error converges in the order of approx. $4\,\mathrm{m}$. Only towards the end of the track, where the geometry regarding the positions of the transmitters relative to the user is unfavorable, the error increases to some extent.

The yellow curve corresponds to the positioning error of DNN-CC-SLAM. The training data for the DNN consists of $8.9\,\mathrm{km}$ of traveled distance from 44 different users. For training and evaluating the DNN, the SFLSE is used as channel estimator.

The positioning error averaged over the entire tracks is $9.0\,\mathrm{m}$ for the single user Channel-SLAM case and $4.2\,\mathrm{m}$ for both cooperative Channel-SLAM and DNN-CC-SLAM. One reason for the good performance of DNN-CC-SLAM is that the DNN is able to average out some positioning errors occurring in cooperative Channel-SLAM during the offline stage. DNN-CC-SLAM, however, does not make use of temporal or spatial correlations, which the particle filter in the Channel-SLAM cases inherently does when tracking the user. The DNN-CC-SLAM error curve is therefore often jumping up and down, i.e., contains relatively many outliers, while the Channel-SLAM error curves are very smooth. Incorporating such correlations at neighboring time instants is part of our ongoing work.

V. CONCLUSION

Traditional fingerprinting schemes suffer from the drawback of the tedious collection of fingerprints at known locations in the offline phase. When the environment changes for example due to new or moved furniture, fingerprints may have to be collected again. The high effort of collecting fingerprints may be a crucial impairment for such schemes.

In our novel approach named DNN-CC-SLAM, we combine a SLAM scheme with fingerprinting. In cooperative Channel-SLAM, multiple users estimate their location based on the channel impulse response using virtual transmitters, and channel parameters are estimated as nuisance parameters. In the first stage of DNN-CC-SLAM, these nuisance parameters and the estimated user positions are used for training a DNN. In the second stage, users can exploit the trained DNN to estimate their position. While cooperative Channel-SLAM suffers from a high computational complexity, estimating a position with the DNN is crucially less complex for three reasons. First, a channel estimator with drastically lower complexity can be used. Second, the Rao-Blackwellized particle filter in Channel-SLAM is replaced by only evaluating the DNN. Third, map matching is not needed. The positioning performance of our proposed method is in the same order of magnitude as the performance of cooperative Channel-SLAM.

ACKNOWLEDGEMENT

This work was partially supported by the DLR project Navigation 4.0.

REFERENCES

[1] K. Witrisal, P. Meissner, E. Leitinger, Y. Shen, C. Gustafson, F. Tufvesson, K. Haneda, D. Dardari, A. F. Molisch, A. Conti, and M. Z. Win, "High-Accuracy Localization for Assisted Living - 5G Systems will turn Multipath Channels from Foe to Friend," *IEEE Signal Process. Mag.*, vol. 33, no. 2, pp. 59–70, Mar. 2016.

[2] H. Wymeersch, N. Garcia, H. Kim, G. Seco-Granados, S. Kim, F. Wen, and M. Fröhle, "5G mm Wave Downlink Vehicular Positioning," in *IEEE Global Communications Conference (GLOBECOM)*, Dec. 2018, pp. 206–212.

[3] R. Mendrzik, F. Meyer, G. Bauch, and M. Win, "Localization, Mapping, and Synchronization in 5G Millimeter Wave Massive MIMO Systems," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2019, pp. 1–5.

[4] M. Ulmschneider, C. Gentner, and A. Dammann, "Cooperative Estimation of Maps of Physical and Virtual Radio Transmitters," in *Proceedings of 34th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2021)*, Sep. 2021.

[5] M. Ulmschneider, C. Genter, and A. Dammann, "Entropy of Transmitter Maps in Cooperative Multipath Assisted Positioning," in *accepted at: IEEE International Conference on Communications (ICC)*, 2022.

[6] T. Jost, W. Wang, U. Fiebig, and F. Perez-Fontan, "Detection and Tracking of Mobile Propagation Channel Paths," *IEEE Trans. Antennas Propag.*, vol. 60, no. 10, pp. 4875–4883, Oct. 2012.

[7] S. He and S. G. Chan, "Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 466–490, 2016.

[8] A. Khalajmehrabadi, N. Gatsis, and D. Akopian, "Modern WLAN Fingerprinting Indoor Positioning Methods and Deployment Challenges," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1974–2002, 2017.

[9] X. Liu, J. Cen, Y. Zhan, and C. Tang, "An Adaptive Fingerprint Database Updating Method for Room Localization," *IEEE Access*, vol. 7, pp. 42 626–42 638, 2019.

[10] Z. Liu, X. Luo, and T. He, "Indoor Positioning System Based on the Improved W-KNN Algorithm," in *IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, March 2017, pp. 1355–1359.

[11] L. Xiao, A. Behboodi, and R. Mathar, “A Deep Learning Approach to Fingerprinting Indoor Localization Solutions,” in *27th International Telecommunication Networks and Applications Conference (ITNAC)*, Nov 2017, pp. 1–7.

[12] K. S. Kim, S. Lee, and K. Huang, “A scalable deep neural network architecture for multi-building and multi-floor indoor localization based on Wi-Fi fingerprinting,” *Big Data Analytics*, vol. 3, no. 1, Apr. 2018.

[13] M. Ulmschneider, S. Zhang, C. Gentner, and A. Dammann, “Multipath Assisted Positioning With Transmitter Visibility Information,” *IEEE Access*, vol. 8, pp. 155 210–155 223, 2020.

[14] M. Ulmschneider and C. Gentner, “RANSAC for Exchanging Maps in Multipath Assisted Positioning,” in *IEEE International Conference on Industrial Cyber Physical Systems (ICPS)*, 2019.

[15] D. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *International Conference on Learning Representations*, 2015.

[16] T. L. Hansen, B. H. Fleury, and B. D. Rao, “Superfast Line Spectral Estimation,” *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2511–2526, May 2018.