

Hybrid Multipath Assisted Positioning and Fingerprinting using Transformer Models

Markus Ulmschneider, Christian Gentner and Armin Dammann *German Aerospace Center (DLR)*

BIOGRAPHY

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 as a researcher at the Institute of Communications and Navigation of the German Aerospace Center (DLR), focusing on multipath assisted positioning. Further research interests are signal processing, sensor fusion and machine learning approaches to localization.

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 has been 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 has been head of the Mobile Radio Transmission research group of the institute.

ABSTRACT

Multipath assisted positioning exploits the spatial information contained in wireless radio signals for localizing a receiver. While such schemes are computationally very complex, they can be used in the offline phase of a fingerprinting scheme. We have presented previously such a fingerprinting scheme, where the fingerprints are channel features, and their locations are estimated with a multipath assisted positioning scheme in the offline phase. The fingerprints and their locations are used to train a mixture density network (MDN). Within this paper, we investigate the benefit of a modern transformer model compared to a MDN.

I. INTRODUCTION

Multipath propagation of wireless signals is traditionally considered harmful to radio localization, since it biases range estimates. However, multipath components (MPCs) contain spatial information regarding the transmitter and receiver location. Hence, they can be exploited for localizing a radio receiver.

One example where multipath propagation can be exploited are fingerprinting schemes. Assuming one or more static transmitters and a mobile receiver, different location-dependent multipath conditions can be mapped in an offline phase and used for positioning in an online phase. Nevertheless, there are also schemes that use multipath propagation in a more native way. In multipath assisted positioning, for example, MPCs are regarded as line-of-sight (LoS) signals from so-called virtual transmitters. Each MPC corresponds to one virtual transmitter, and the LoS component, if present, to the physical transmitter. These virtual transmitters can be used for localization of the receiver just like the physical transmitters. In the following, the term transmitter comprises both physical and virtual transmitters. Physical transmitters could be WiFi routers or cellular base stations, for example. In the same way, the term signal component comprises both the LoS component and MPCs.

Typically, the locations of both the virtual and the physical transmitters are unknown. Nevertheless, these transmitter locations can be estimated jointly with the receiver position with simultaneous localization and mapping (SLAM) as in Durrant-Whyte and Bailey (2006). There exist various such multipath-based SLAM algorithms in the literature, such as in Witrisal et al. (2016); Leitinger et al. (2019); Wymeersch et al. (2018); Mendrzik et al. (2019). We have previously introduced such an approach named Channel-SLAM, for which a single user version presented in Gentner et al. (2016) and a cooperative version presented in Ulmschneider et al. (2021) exist.

Multipath assisted positioning schemes tend to be computationally very expensive due to complicated signal processing algorithms used. A realization of such schemes in real world scenarios is therefore often times not feasible. Nevertheless, they can be used in fingerprinting to drastically reduce the physical effort in the offline phase. We have recently presented a scheme that uses cooperative Channel-SLAM in the offline phase of a fingerprinting scheme in Ulmschneider et al. (2022); Ulmschneider and Gentner (2023). We name this scheme DNN-CC-SLAM for convenience. In the offline phase of DNN-CC-SLAM, the locations of the fingerprints are collected with cooperative Channel-SLAM. The fingerprints themselves are delays of signal components. In this way, no 3rd party positioning system needs to be deployed to obtain the fingerprint locations, and the delays of signal components are even nuisance parameters estimated in cooperative Channel-SLAM. A deep neural network (DNN)

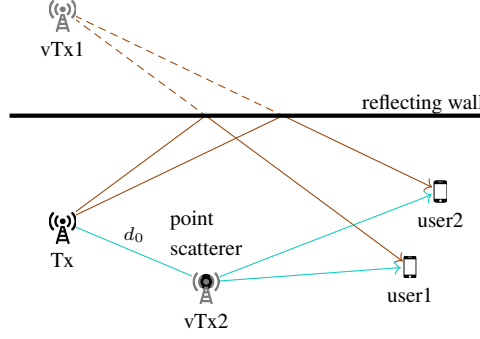


Figure 1: The fundamental idea behind multipath assisted positioning is to treat every MPC as a LoS signal from a virtual transmitter.

is then trained to learn receiver locations based on fingerprints. In the online phase, the DNN is evaluated using measured fingerprints to obtain a location. Passing these locations through a downstream Kalman Filter as in Kalman (1960) at last drastically improves the positioning performance.

Within this paper, we want to further analyze and improve DNN-CC-SLAM. In the online phase, very efficient channel estimators can be used, and the Kalman filter is also very efficient. Nevertheless, we aim at omitting these two operations and instead use a transformer. Transformer models were introduced in Vaswani et al. (2017) and have recently gained a lot of attention due to their huge successes in large language models. However, they have also successfully been applied to various time series tasks as presented in Wen et al. (2023).

Transformers take the temporal and spatial correlations regarding the receiver positions into account. Therefore, we omit the Kalman filter in the transformer case. In addition, we want to feed the transformer with the correlation function with respect to the received and the transmitted signal. Hence, we do not require channel estimation in the online phase. We analyze the performance a transformer in DNN-CC-SLAM in comparison with standard machine learning approaches such as feedforward neural networks or MDNs.

The remainder of this paper is organized as follows. In Section II, we introduce the concepts of cooperative Channel-SLAM, fingerprinting and DNN-CC-SLAM. Section III presents the transformer and its use in DNN-CC-SLAM. We perform simulations to evaluate the transformer in Section IV, and conclude the paper in Section V.

II. MULTIPATH ASSISTED POSITIONING AND FINGERPRINTING

The idea of multipath assisted positioning is illustrated in Fig. 1, where the transmit signal from the physical transmitter arrives at the receiver via different propagation path, omitting the LoS path for clarity. The MPC reflected at the wall is interpreted as a LoS signal from the virtual transmitter $vTx1$. The location of this virtual transmitter is at the location of the of the physical transmitter mirrored at the wall. Likewise, the MPC scattered at the point scatterer is interpreted as a LoS signal from the virtual transmitter $vTx2$, which is located at the point scatterer. Multiple interactions of the transmit signal with different reflective surfaces or point scatterers can be modeled in a straightforward manner.

Assuming that the MPCs originate from single or multiple interactions with straight walls or point scatterers, the locations of these virtual transmitters are independent from the receiver position. Hence, the locations and further properties of the virtual transmitter, such as delay offsets to the respective physical transmitter, can be estimated jointly with the user position. We have previously introduced such a scheme named Channel-SLAM, where this joint estimation is performed with SLAM. In order to improve the positioning performance and convergence time, we have also come up with cooperative Channel-SLAM, where multiple receivers exchange information regarding the states of physical and virtual transmitters.

In the first step of Channel-SLAM, a channel estimator estimated and tracks parameters of signal components over time. In the second step, these estimates are used in a particle filter to simultaneously localize the receiver and map the transmitters with SLAM. In cooperative Channel-SLAM, different receivers exchange information on the transmitter locations. With prior information on transmitter locations, the positioning performance improves drastically, especially in the long term.

While multiple such multipath assisted positioning-based SLAM schemes have emerged over the last years, they have in common that they suffer from a very high computational complexity due to complex signal processing. This high complexity makes a real-world deployment unfeasible for many use cases. Nevertheless, such schemes can be used in the offline phase of a fingerprinting scheme. We have previously introduced such a scheme named DNN-CC-SLAM, where cooperative Channel-SLAM is used in the offline phase to collect the fingerprints and estimate the corresponding locations.

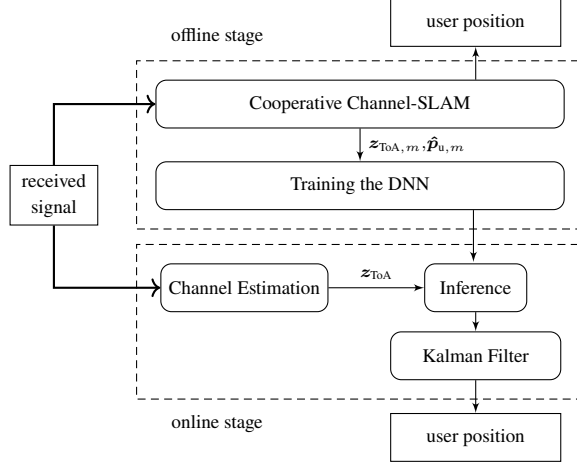


Figure 2: A flow chart of DNN-CC-SLAM.

An overview of DNN-CC-SLAM is presented in Fig. 2. In the offline phase, fingerprints are collected and their locations estimated with cooperative Channel-SLAM. fingerprints are the delays of signal components, which are nuisance parameters from the channel estimator in cooperative Channel-SLAM. A DNN is then trained with these data to predict the receiver location given the channel features, i.e., delays of signal components. In particular, we use a MDN to not only predict a location, but a normal distribution of the receiver location. In the online phase, efficient channel estimators can be used to extract the delays of signal components, which are passed to the DNN for inference. Using a MDN, the resulting distribution is fed into a Kalman filter as measurement, tracking the receiver over time.

The advantage of DNN-CC-SLAM is twofold. On the one hand, the physical effort to collect the fingerprints and in particular their locations is drastically lower than for standard fingerprinting schemes. In such standard schemes, a third-party localization system typically needs to be deployed to obtain the locations of measured fingerprints, which is not necessary in our case. Even more, in case the environment changes and new fingerprints need to be collected, these third-party systems have to be deployed again. On the other hand, while the computational complexity is very high in the offline phase, it is very low in the blocks in the online phase in Fig. 2, i.e., when using the possibly constrained receiver hardware: very efficient channel estimators can be used, since no array processing or parameter tracking are necessary. Inference of a neural network can be implemented very efficiently, in particular on modern computer or smartphone chipsets. Finally, the Kalman filter is very efficient, since the MDN 'measures' the receiver position, which is already part of the state space.

III. TRANSFORMER

While all blocks in the online phase of DNN-CC-SLAM are efficient, we seek to investigate more modern machine learning approaches within this paper. In particular, we want to investigate, how a modern transformer model can be used in DNN-CC-SLAM.

Transformers have not only revolutionized modern large language models (LLMs), but have found their way into various time series processing applications. Compared to more traditional machine learning approaches, their strength lies in their ability to model long-range dependencies efficiently. The key component of transformers is the attention mechanism, which is responsible for capturing the relationships between different elements in a sequence. It allows each element to dynamically focus on relevant elements within the sequence, based on their importance or relevance.

While various variants of the original transformer architecture have been proposed, the patch time series transformer (PatchTST) from Nie et al. (2023) adapts transformers for time series analysis by employing patch-based processing, where sub-sequences of the time series are treated as input tokens to the transformer, enabling efficient modeling of long-range dependencies and temporal patterns.

We employ such a PatchTST architecture for DNN-CC-SLAM. Compared to our DNN-CC-SLAM flowchart in Fig. 2, we make the following notable changes. First, the DNN is replaced by a PatchTST. Second, the fingerprints learned by the PatchTST are not a list of delays of signal components, but the correlation function with respect to the received and the transmitted signal. Hence, the channel estimator in the online phase is not needed any more. Third, since transformer models are able to capture temporal dependencies already, we omit the Kalman filter in the online phase. Consequently, in the online phase, only the inference block stays present.

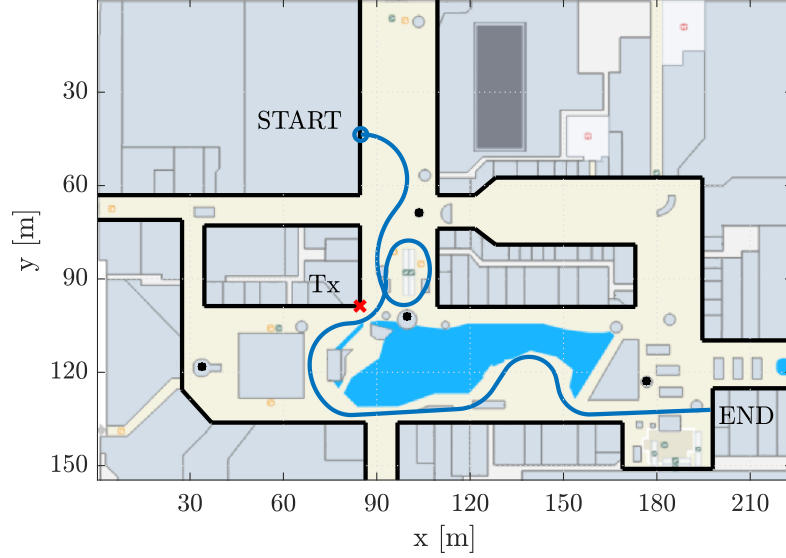


Figure 3: Top view of the simulation scenario with one physical transmitter labeled Tx and a receiver trajectory in blue.

Looking at our implementation of the transformer, the time series used as fingerprints are the correlation functions mentioned above. In the PatchTST, each correlation function is split into single patches, which may or may not overlap. The output of the transformer is a two-dimensional vector indicating the receiver location.

IV. SIMULATIONS

1. Simulation Scenario and Parameters

Our approach is evaluated with simulations in the scenario presented in Fig. 3, which shows the top view of an indoor mall. There is one single physical transmitter labeled Tx, depicted by the red triangle. This transmitter continuously transmits a signal that is rectangular in frequency domain. The signal has a bandwidth of 100 MHz at a center frequency 1.9 GHz. The multipath propagation is defined by the walls and scatterers, depicted by the black lines and dots, respectively. Based on these walls and scatterers, the true channel impulse response (CIR) is obtained with ray-tracing for every receiver position. At the receiver, a snapshot of the received signal is recorded every 100 ms, leading to an update rate of 10 Hz. In Fig. 3, there is one receiver track running from START to END, which we will use to evaluate the positioning performance.

For Channel-SLAM, i.e., in the offline phase of DNN-CC-SLAM, each receiver is equipped with an antenna array of nine elements arranged uniformly in a 3×3 grid. In the particle filter of Channel-SLAM, the time of arrival (ToA) and angle of arrival (AoA) estimates are used as measurements. While turn rates from a gyroscope that is rigidly mounted to the receiver and thus aligned with the receiver heading are used as control input in the particle filter in Channel-SLAM, no such additional sensors are used in the online phase of DNN-CC-SLAM. Hence, no array processing is necessary in this online phase.

The number of samples in the correlation function is 101. In case of using a standard feedforward DNN or MDN in DNN-CC-SLAM, this is also the size of the network's input layer. The sizes of the hidden layers are 1000, 300, 200, 100, 50 and 30 neurons. The output layer size depends on the number of Gaussian components in the MDN. For our evaluations, we chose two components following Ulmschneider et al. (2024), leading to an output layer of size twelve.

For the transformer, the input sequence length is 101 as well, corresponding to the correlation function length. The patch size and stride length are 48 and eight, respectively. The model dimension is 128, and there are three stacked encoder/decoder layers.

In our evaluations, we obtain training data from 44 different user tracks with a combined traveled distance of approx. 8.86 km. The training data are correlation functions and estimated receiver positions obtained with cooperative Channel-SLAM as indicated in Fig. 2. Given a constant speed of 1 m/s, there are 88,640 data points, of which 90% are used for training and 10% as validation data.

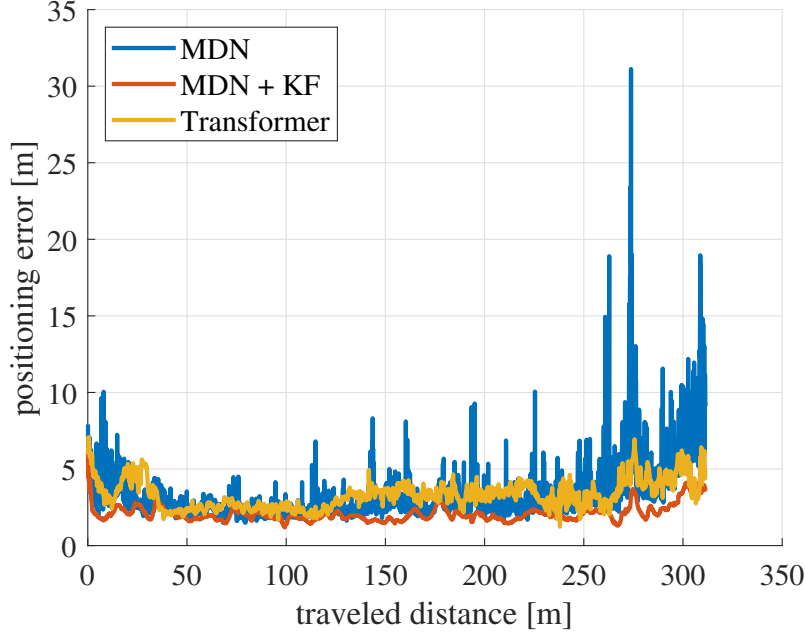


Figure 4: Positioning performance of DNN-CC-SLAM.

2. Simulation Results

The positioning performances of DNN-CC-SLAM are depicted in Fig. 4 for the track depicted in Fig. 3. This track was neither used for training nor for validation. Due to the stochastic nature of the involved methods, all results are averaged over 50 runs. Due to the long training time, the results for the transformer are averaged over 10 runs.

The blue curve labeled “MDN” shows the positioning error using a MDN with one single component without a Kalman filter. The red curve labeled “MDN + KF” shows the positioning error using a MDN and a subsequent Kalman filter. We can see that the Kalman filter drastically improves the positioning performance, taking into account temporal and spatial correlations, which the MDN itself is unable to capture by nature.

The positioning error of the PatchTST is plotted orange and labeled “Transformer” in Fig. 4. We can see that the transformer performs better than the MDN without a Kalman filter. However, adding the Kalman filter to the MDN performs better than the transformer. In our case, it turns out that a simpler machine learning model, i.e., the MDN, together with an optimal filter under the Gaussian assumption performs better than the transformer with its rather complicated architecture.

3. Timing

In the following, we want to practically evaluate the methods that we have analyzed in terms of computational complexity. Our hardware platform is a Intel[®] Xeon[®] processor E5-2697 with a maximum frequency of 2.3 GHz, on which we will compare the actual execution time in Python for all machine learning methods. However, this part should be addressed with care for the following reasons. First, our implementations use different backends, with possibly different optimization levels. Second, while we try to take care of efficient implementations, there may be more efficient ways of implementing certain functionalities. Third, some backends may be optimized for certain hardware. The relative results may therefore be different on a graphics processing unit (GPU), for example. This includes parallelization capabilities of some backend algorithms. Fourth, a careful hyperparameter tuning is crucial for not only the performance, but also the computational complexity of a method.

Nevertheless, we want to present our results to give at least an impression on the computational complexities, in particular on the relative complexities of the MDN and the transformer approach. Looking at the flow chart in Fig. 2, we focus on training the DNN in the offline phase and the different methods in the online phase.

The channel estimator superfast line spectral estimation (SFLSE) in the online phase is the most complex block in the online phase, taking on average 195.4 ms of processing time at each time instant along the track in Fig. 3. It is implemented in MATLAB[®]. Using the correlation function as input to the DNN as described in Section III, this step can be skipped without notable changes in the performance.

The Kalman filter is implemented using the numpy library in Python. The processing time of the Kalman filter is 0.383 ms at each time instant. The inference time in a MDN with two components is even significantly shorter with 0.083 ms. We attribute that to a very efficient inference implementation of the Tensorflow framework and the simplicity of the fully connected MDN.

Regarding the transformer, the inference time is 0.028 ms, which is even more efficient than the fully connected MDN. We have implemented a PatchTST from the Huggingface platform presented in Wolf et al. (2019). The number of trainable parameters is 601,346 for the PatchTST compared to 488,712 parameters in the MDN.

We again want to emphasize that our complexity evaluations are highly dependent on the used libraries and backends, on the choice of hyperparameters, the efficiency of own implementations and of course on the specific hardware used.

V. CONCLUSION

In cooperative Channel-SLAM, multipath propagation is exploited for localizing a receiver. While multipath assisted positioning schemes in general are computationally very complex, we use cooperative Channel-SLAM in the offline phase of a fingerprinting scheme named DNN-CC-SLAM in order to decrease the physical effort for collecting fingerprints drastically.

We have analyzed a transformer model in DNN-CC-SLAM, compared to our standard MDN. We have seen that the transformer can not compete with the MDN and a downstream Kalman filter in terms of positioning performance. Nevertheless, our evaluations indicate that the transformer is more computationally efficient in our implementation.

REFERENCES

- Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous Localization and Mapping: Part I. *IEEE Robot. Autom. Mag.*, 13(2):99–110.
- Gentner, C., Jost, T., Wang, W., Zhang, S., Dammann, A., and Fiebig, U.-C. (2016). Multipath Assisted Positioning with Simultaneous Localization and Mapping. *IEEE Trans. Wireless Commun.*, 15(9):6104–6117.
- Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45.
- Leitinger, E., Meyer, F., Hlawatsch, F., Witrisal, K., Tufvesson, F., and Win, M. Z. (2019). A Belief Propagation Algorithm for Multipath-Based SLAM. *IEEE Trans. Wireless Commun.*, 18(12):5613–5629.
- Mendrzik, R., Wymeersch, H., Bauch, G., and Abu-Shaban, Z. (2019). Harnessing NLOS Components for Position and Orientation Estimation in 5G Millimeter Wave MIMO. *IEEE Trans. Wireless Commun.*, 18(1):93–107.
- Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. (2023). A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *The Eleventh International Conference on Learning Representations*.
- Ulmschneider, M. and Gentner, C. (2023). User Tracking with Multipath Assisted Positioning-based Fingerprinting and Deep Learning. In *17th European Conference on Antennas and Propagation (EuCAP)*.
- Ulmschneider, M., Gentner, C., and Dammann, A. (2021). Cooperative Estimation of Maps of Physical and Virtual Radio Transmitters. In *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*.
- Ulmschneider, M., Gentner, C., and Dammann, A. (2022). Learning-Based Fusion of Multipath Assisted Positioning and Fingerprinting. In *Proceedings of 35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2022)*.
- Ulmschneider, M., Gentner, C., and Dammann, A. (2024). Mixture Density Networks for Multipath Assisted Positioning-based Fingerprinting. In *18th European Conference on Antennas and Propagation (EuCAP) 2024*.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is All you Need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., and Sun, L. (2023). Transformers in Time Series: A Survey. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 6778–6786. International Joint Conferences on Artificial Intelligence Organization. Survey Track.
- Witrisal, K., Meissner, P., Leitinger, E., Shen, Y., Gustafson, C., Tufvesson, F., Haneda, K., Dardari, D., Molisch, A. F., Conti, A., and Win, M. Z. (2016). High-Accuracy Localization for Assisted Living - 5G Systems will turn Multipath Channels from Foe to Friend. *IEEE Signal Process. Mag.*, 33(2):59–70.

- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2019). HuggingFace's Transformers: State-of-the-art Natural Language Processing.
- Wymeersch, H., Garcia, N., Kim, H., Seco-Granados, G., Kim, S., Wen, F., and Fröhle, M. (2018). 5G mm Wave Downlink Vehicular Positioning. In *IEEE Global Communications Conference (GLOBECOM)*, pages 206–212.