Two Staged ANN Based UWB Ranging Error Mitigation for Real Time Self Localization on Mobile Robots

Steffen Schmelter[a] and Christof Röhrig[a]
[a]Institute for the Digital Transformation of Application and Living Domains (IDiAL), Otto-Hahn-Str. 23, 44227 Dortmund, Germany. Email: steffen.schmelter@fh-dortmund.de, christof.roehrig@fh-dortmund.de

Abstract

Ultra-wideband (UWB) offers global positioning in various, mostly indoor scenarios, whereby the localization can be performed by the device itself. The usual accuracy of its proximity detection under good conditions is ± 10 cm. However, this accuracy can be strongly influenced by environmental influences resulting in highly inaccurate and unsatisfactory position estimation. This paper introduces a method utilizing artificial neural networks (ANNs) to mitigate errors by detecting non-line-of-sight (NLOS) conditions and correcting measured distances. Two ANNs are created, which are capable to exploit information using both the additional metrics such as the channel impulse response (CIR) provided by the Decawave DW1000 UWB radio chip.

1 Introduction

1.1 Motivation

UWB is a popular technology for wireless localization in various, primary indoor scenarios where position of mobile robots or systems must be determined by the device itself. Compared to other localization technologies, like e.g. laser scanners, UWB based localization tends to be less expensive whilst also offering global positioning. Under usual line-of-sight (LOS) conditions, UWB offers a proximity detection to an accuracy of ± 10 cm using two-way ranging time-of-flight (TOF) measurements [1]. However, the general localization accuracy is highly dependent on the environment and environmental influences. Ranging errors can be caused by refractions, reflections, multipath effects, blockings in signal path and the general complexity of the indoor radio environment. Leading to a positive bias in the ranging measurement, those errors can easily result in inaccurate and unsatisfactory position estimation. There are several approaches on ranging error mitigation with conventional methods [2][3]. Recent publications show that exploiting of ANNs is a promising solution to solve these problems in a more effective and efficient way [4][5][6][7][8][9].

1.2 Related Work

Following recent work and literature, the solving strategy for UWB ranging errors can be coarsely broken up into two steps [10]: (1) A process in which is decided if a ranging classifies as either LOS or NLOS or multi path (MP) as a third class. Followed by a (2) mitigation procedure, in which a bias is applied to correct the ranging in some way. Depending on the algorithm used for position estimation (e.g. a Kalman filter), rangings which previously have been identified as NLOS have to be either discarded or also being corrected during this process. Commercial UWB radio IC's like the Decawave DW1000 provide additional values and ranging metrics stored in externally accessible registers for each measurement. Once, there is a channel impulse response (CIR) register containing 1016 values consisting out of a 16-bit real and a 16-bit imaginary part. Furthermore, there are single value registers with metrics such as registers containing on-chip calculated metrics based on the CIR. In [4] an approach is described using machine learning (ML) to exploit the mentioned single value registers and on-chip metrics to predict whether a ranging is either LOS, NLOS or MP. Other work [5] uses the same registers, but to predict individual biases for error mitigation. Limiting to the single value registers and on-chip metrics as only input to an ML-algorithm has the advantage, that only a few bytes need to be fetched from the UWB radio IC, but also more detailed information is missed out therefore. The CIR has a size of 4046 bytes in total, to reduce the amount of transmitted data, algorithms such as the following only use a partial cut-out of the CIR. To classify if a ranging is a LOS or NLOS condition [6][7] uses only 120 values from CIR. The same applies to the approaches [8][9] where maximum cutout-sizes of 128 respectively 160 are fed into neural networks used to mitigate ranging errors.

This Project is supported by the Federal Ministry for Economic Affairs and Climate Action (BMWK) on the basis of a decision by the German Bundestag, grant No. KK5119001BD0

1.3 Approach

This paper introduces an approach featuring two ANNs which could be used to mitigate ranging errors when using UWB for real time self localization. Both ANNs share are a nearly identical architecture, but each network model takes on a different task. The first model predicts whether a ranging classifies as a LOS or NLOS condition. The second is used to compensate the remaining error in all rang-

ISBN 978-3-8007-5891-3

ings identified as LOS. The models utilize both the chip's provided single value registers and on-chip metrics as also the CIR by featuring an input divided into two branches. Other work has shown that UWB rangings have specific a bias correlating with the distance [2][11]. Further, as mentioned in the previous section, there are approaches using either only the on-chip metrics or only the CIR. For this reason, it is obvious to examine if further potential can be exploited by utilizing both data in combination.

2 Dataset construction

2.1 Data collection procedure

Data collection was performed through two different approaches, in a manner hereinafter referred to as static and dynamic. The anchors and tag used in the data collection process were MDEK1001 boards, featuring a Decawave DW1000 UWB radio chip. Overall 5 individual anchors (referred to as A1-A5) and one tag (T1) were used. To compensate warm-up-drift, the whole system was allowed to acclimate first and afterwards dry-runned for 15 minutes, until measurements were taken.

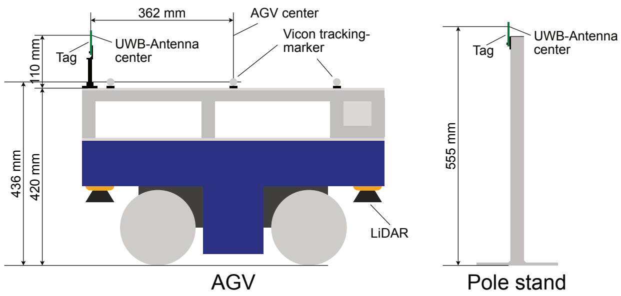

Figure 1 Schematic overview of the mounting positions of the UWB Tag (T1) on the AGV (dynamic) and the pole stand (static data collection approach).

For the static approach, the UWB tag (T1) and anchor (A5) were both mounted each on a metal pole stand, as schematically shown in the right part of **Fig. 1**. For the dynamic approach, the T1 was mounted on an AGV (see Fig. 1, left) and anchors A1-A4 were placed with 45-degree arrangement in the room corners (see **Fig. 3**). For ground truth reference, the real distances between anchors and tag were determined with either a tape measure (static approach) or a 3D tracking system (Vicon). In case of the static approach, 5 scenarios with various ambient conditions were created in total (see **Tab. 1**). In all scenarios, the distance between A5 and T1 was increased in either 0.5 or 1.0 m increments by moving A5 while T1 remained on a static position all time. On each increment, 100 measurements were triggered under LOS and NLOS condition each. Except for the datasets "Office" and "Lab.", the measurements were carried out over a distance range of 0.5 to 26.0 meters. To provoke a NLOS condition, a solid 3 mm thick metal plate was placed with an approx. spacing of 40 cm in front of T1. Note that some sceanrios feature additional measurements for LOS respectively no NLOS measurements at all. Furthermore, in some cases the data received from the

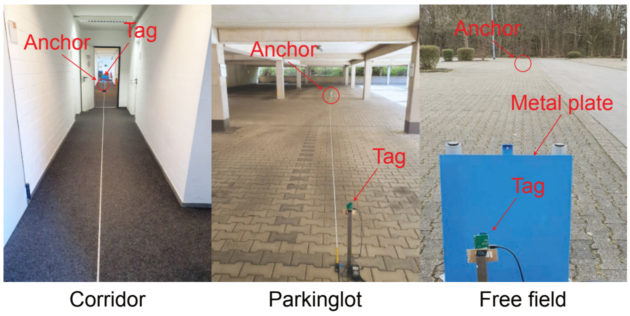

Figure 2 Static approach: images from three experimental setups.

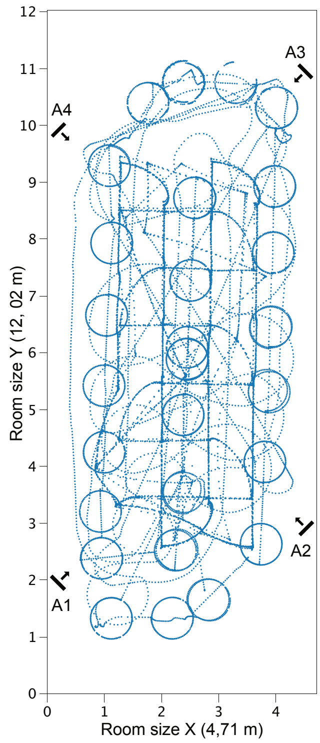

Figure 3 Plot of all ground truth positions in the "Lab." room on which a measurement to one of the anchors (A1-A4) was initiated by T1. Ground truth was determined via Vicon.

MDEK1001 was fragmented, those measurements were dropped. This results in the smaller number of samples ultimately collected, compared to the number that would have to result from the calculation. In total 24,849 samples were recorded whereby 14,652 are LOS and 10,197 are NLOS. **Figure 2** shows images of some static scenario setups. For gathering dynamic data, the AGV autonomously followed various courses but was also driven manual. All rides were carried out in a mostly empty room ("Lab") with a size of 4.71 by 12.02 m. In Fig. 3, all ground truth positions, on which a measurement to one of the anchors (A1-A4) was initiated by T1, are plotted. To provoke a NLOS condition, during some measurements one of each anchor was shielded with the mentioned metal plate. In total 15,866 samples were recorded whereby 14,754 are LOS and 1,112 are NLOS (see **Tab. 2**).

		Std.	MAE	Min.	Max.
Free field	L	5.11	26.81	5.6	40.13
L: 2,574	N	5.85	39.34	19.87	65.10
N: 2,574	A	8.33	33.07	5.6	65.10
Parking lot	L	8.92	28.67	6.94	110.29
L: 2,574	N	18.72	49.70	13.88	183.45
N: 2,574	A	18.04	39.18	6.94	110.29
Corridor	L	14.38	38.47	-4.15	122.77
L: 7,722	N	26.73	64.01	9.44	181.04
N: 5,049	A	23.74	48.57	-4.15	181.94
Office	L	8.88	44.17	24.97	96.44
L: 693					
Lab.	L	5.88	36.29	13.79	51.68
L: 1,089					

Table 1 Static scenario: statistics for uncalibrated distance errors (cm) (total number of samples per class, standard deviation, mean absolute error (MAE), minimum, maximum) for measured UWB distances (T1 to A5). L = LOS, N = NLOS, A = All (LOS ∪ NLOS).

		Std.	MAE	Min.	Max.
A1	L	13.21	34.17	-19.55	121.05
L: 3,689	N	21.92	64.88	24.37	128.96
N: 285	A	16.11	36.38	-19.55	128.69
A2	L	10.42	27.28	-22.44	146.56
L: 3,691	N	18.00	68.98	32.05	136.54
N: 275	A	15.37	30.17	-22.44	146.56
A2	L	16.96	37.75	-2.71	132.68
L: 3,684	N	19.17	60.02	17.19	145.65
N: 283	A	18.06	39.34	-2.71	145.65
A4	L	11.69	29.93	-48.47	110.20
L: 3,690	N	22.23	61.15	22.01	136.76
N: 269	A	14.92	32.05	-48.47	136.76

Table 2 Dynamic scenario: statistics for uncalibrated distance errors (cm) for measured UWB distances of T1 to A1-A4. All measurements were performed in room "Lab." of Fig. 3

2.2 Data structure

The used DW1000 chip offers several single value and on-chip metrics such as the CIR register, which contains 1016 sample values, consisting out of a 16-bit real and a 16-bit imaginary part each (4064 bytes in total). In **Tab. 3** an overview is given about all the chip's provided values that were captured with each measurement. To reduce the amount of transferred data, the CIR was cropped to retain 128 samples only, starting with an offset of -20 samples relative to the first path index (FP index) (see **Fig. 4**). In terms of the radio signal's TOF, one CIR sample equals 1 nanosecond or 30 cm [1]. Note that the 16-bit on-chip value of the FP index is divided into a 10-bit whole part (most significant bits) and a 6-bit fractional part. The fractional part can be used to determine a more accurate position of the FP index in between two CIR samples. In this case the 6-bit fractional part was only used to round up or down the reported 10-bit coarse position. Additionally, it should be noted that the resolution of distance reported by the chip is 1.8761 cm.

	Bits	Values	Type
Distance (m)	64	1	Double
FP Index	16	1	unsigned Integer
FP Amp 1	16	1	unsigned Integer
FP Amp 2	16	1	unsigned Integer
FP Amp 3	16	1	unsigned Integer
Max Grow CIR	16	1	unsigned Integer
Max Noise	16	1	unsigned Integer
RX Pre Count	16	1	unsigned Integer
Std Noise	16	1	unsigned Integer
CIR real	16	128	Integer
CIR imaginary	16	128	Integer
Total Bytes	534		

Table 3 Structure of the data captured from the DW1000 with each UWB measurement.

The plots of absolute values from the CIR shown in Fig. 4, already indicate a trend that there are differences in the CIR under LOS and NLOS conditions. Under LOS conditions, measurements seem to share a similar waveform, and their highest peak mostly seems to begin at the reported FP index. In the NLOS case, the signal travels along many routes until it reaches the antenna which makes the TOA estimation ambiguous.

The selected crop size of the CIR seems to be sufficient, since the signal has mostly decayed by the end of the section. At the time the firmware was developed, it was not known to what extent signal peaks may occur ahead of the FP index. Hence, the offset was chosen that large. Since there are no outliers larger than -7 across all measurements at all, the offset may be reduced in future work. The reduction can be either used to reduce the total number of transmitted samples or to include more samples after the FP index while keeping the same crop size.

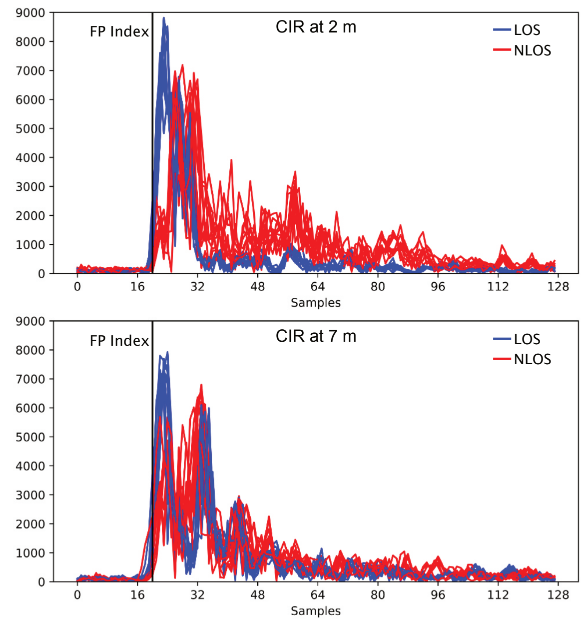

Figure 4 Plot of absolute (raw, non normalized) values from the CIR under LOS and NLOS conditions from the "Corridor" dataset at a static 2 and 7 m distance. Both plots unify 10 CIR's each per LOS and NLOS condition.

2.3 Dataset creation

Based on the collected data, two datasets for training and testing the networks were created. Both datasets contain data from static such as dynamic measurements; the exact composition is shown in **Tab. 5**. The static part of the training-set includes all measurements from the "Parking lot" and "Corridor" datasets. The dynamic part contains 12,632 measurements from some autonomous, such as all manual driven courses of the AGV. The test-set's dynamic and static portion consist out of all the remaining dynamic data, that was not used for the training-set.

To prepare the data for use with machine learning, all values are normalized to a value range between 0 and 1. Regarding the real and imaginary values from the CIR register, which also contain negative values, the values have been scaled and shifted in a way so that $0 \hat{=} 0.5$. Furthermore, all distance measurements from anchors (A1-A5) were corrected with an individual static offset value (bias) (see **Tab. 4**) so that their distance error becomes more uniform.

A1	A2	A3	A4	A5
-31.9	-25.0	-33.0	-27.7	-33.5

Table 4 Static calibration offset values (bias) in cm that was applied for each anchor.

	LOS	NLOS	ALL
TRAIN			
Dynamic	12,072	560	12,632
Static	10,296	7,623	17,919
All	**73.2% / 22,368**	**26.8% / 8,183**	**30,551**
TEST			
Dynamic	2,682	552	3,234
Lab.	1,089	-	1,089
Office	693	-	693
Free field	2,574	2,574	5,148
All	**69.2% / 7,038**	**30.8% / 3,126**	**10,164**

Table 5 Composition of the number of measurement samples per dataset.

3 Approach

3.1 Network design

In **Fig. 5** an overview of the overall network architecture is presented. Note that two separate networks where created. Apart from the output layer, both networks share an identical architecture featuring an input consisting out of two separate branches: (1) a 2×128 sized matrix such as (2) a one-dimensional input field of 9 values. The input branch (1) is used to process the real and imaginary values and consists out of one 2D convolutional and two 1D consecutive convolutional layers followed by a fully connected stage. By using a 2D convolution in the first layer, it is intended to keep a correlation between real imaginary part of the values from the CIR. With each progression in the convolutional layer depth, the number of filters (f) is rose from 24 to 48, the kernel size (k) is decreased from 2×10 to 3 and the stride (s) increased from 1 to 2. Input branch (2) is used to process the mentioned on-chip metric provided values and is built out of four decreasing consecutive fully connected layers. Both branches result in an output of 8 features each, which are concatenated and fed into a last fully connected stage of two layers. All layers, except for the output layer, use a Leaky-ReLu activation function with $\alpha = 0.1$. The output (c) of the first network, used for classification, consists out of two neurons and uses a softmax activation function, and no bias is applied. Output (r) of the second network used for error mitigation, is a single value and has no activation function (linear), but uses a static bias of -0.2. The first network is trained to perform a binary classification to identify whether a UWB ranging is a LOS or NLOS condition. Each output holds a predicted probability for its corresponding class. This computed probability may be used to compute a covariance to input into a Kalman filter. Mainly it is used to filter out all measurements which are classified as NLOS at the moment. All LOS measurements are further processed with the second network, which is trained to predict a value, representing the ranging error (regression). The predicted ranging error is then subtracted from the UWB distance.

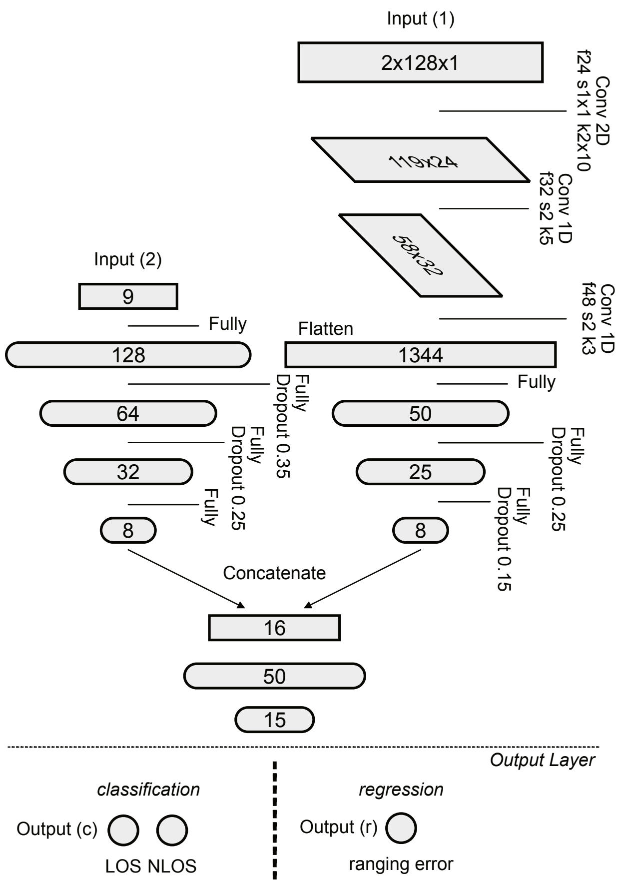

Figure 5 Architectural overview of the neural network model. Rounded rectangles represent fully-connected layers. The classification model has 91,290, the regression model 91,276 (trainable) parameters in total.

3.2 Training

The Adam optimization algorithm [12] is employed for training of both networks. The momentum parameters are set as $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-8}$. The networks are trained separately with the three datasets from Tab. 5 ("Dynamic", "Static" or "All").

3.2.1 NLOS detection (classification)

In case of training the classification network, the learning rate is set to 0.001 and is kept constant for 20 epochs with a batch size of 128 using the sparse categorical crossentropy loss function.
For the training-set such as the test-set, the whole data is shuffled and the occurrences of LOS and NLOS samples are equalized. As the number of occurring NLOS samples in the datasets is lower than that of the LOS, especially in the dynamic case, the total number of samples usable for training is reduced drastically. In the dynamic part of the training-set, for example, the number of NLOS samples is 560. To reach a balanced distribution, only 560 LOS samples are kept, while the remaining have to be discarded, resulting in a training-set of only 1120 samples in total.

3.2.2 Error mitigation (regression)

The regression model is trained for overall 40 epochs. The learning rate is 0.001 for 20 epochs and then reduced to 0.0001 for another 20 epochs. As loss function the mean squared error (MSE) is applied.
Due to the used Leaky-ReLu activation function, the network tends to eliminate or at least distort values below zero. Therefore, the preferred scale for the network's output value should be in the 0 to 1 range. To get the ranging error as a label, the ground truth distance is subtracted from the raw UWB distance. Following Tab. 1 and 2, except for some outliers, the majority of errors is located in the cm range. Furthermore, errors in the raw UWB distance tend to have a mostly positive bias when compared to the ground truth. Since ground truth and raw UWB distance are given in meter (m), this will (mostly) result in the preferred scale. Due to the previously applied offset calibration (see Tab. 4) in some cases negative correctional biases appear. To finally facilitate the output of also negative values by the network, the output layer is provided with the previously mentioned static bias of -0.2. Since NLOS measurements are intended to be identified and filtered out utilizing the classification network as a previous stage, all NLOS data is excluded from the training and test datasets.

3.3 Experimental results

3.3.1 Performance evaluation

Figure 6 shows the confusion matrices for the classification model when trained with data from the training-set in three different combinations. Each separately trained model performs classification on three different test-set combinations. Note that the datasets "Lab" and "Office" (see Tab. 5) are left out as they do not contain any NLOS samples; the "All" dataset therefore only includes data from "Dynamic" and "Free Field" in this case.

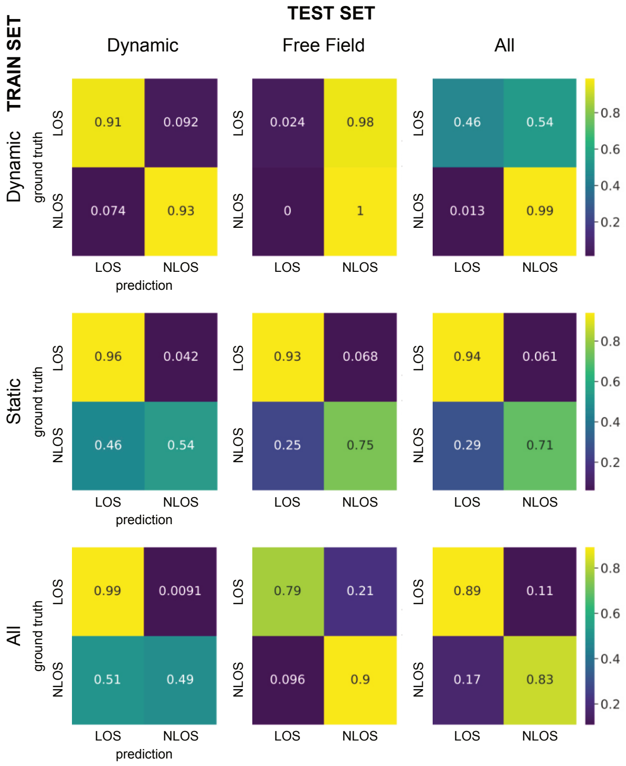

Figure 6 Cross-validation results for LOS / NLOS classification results on different test-sets.

As to be expected, the network trained with data from one specific domain (dynamic or static) performs best when tested on data similar to the data it was trained with. Whereby the network trained on static data only, tends to have better performance overall compared to the one trained with dynamic data. However, this must be put into perspective, considering that the size of the training set containing dynamic data is only 7.3% the size of the static. The same applies to the test-set, whereby it corresponds to 21.4% in this case. Furthermore, it has to be noted that the portion of data included from the dynamic dataset is larger in the "All" dataset. As mentioned in previous section 3.2.1, both datasets are merged and equalization takes place only afterwards by picking samples randomly until an equal distribution is reached. Thus explaining the nearly 50/50 classification on the "All" dataset, when the network is trained with the "Dynamic" data only (top right corner in Fig. 6).
In **Tab. 6** the results of the range error mitigation model are shown. The three individually trained models are evaluated on the four single test-sets such as the overall dataset combining all the first three datasets from indoor environments. In general, the model trained on dynamic data only, seems to outperform the model trained with just the static data - in this case, the dynamic and the static datasets share a more similar sizing and therefore better comparability. Even compared to the model trained on all data, the first model achieves a better performance. It is able to reduce the standard deviation such as the mean absolute error (MAE) on all datasets, except for "Lab" and "Free Field", but also reaches the best performance on "All". However, it

ISBN 978-3-8007-5891-3

has to be taken into account that the amount of static measurement data from the "Lab" and "Office" datasets have a smaller share in the "All" dataset. Furthermore, the static portion of data in the training-set originates from scenarios with very specific environmental conditions: a narrow, long corridor and a concrete parking lot without delimitation by walls. When considering standard deviation only, the model trained on the "All" dataset has the (almost) relatively lowest values on all five test-set's, compared to the two other models. Overall, however, it also tends to a deterioration on datasets "Lab" and "Free Field" compared to raw UWB.

		Dyn.	Lab.	Office	All	Free Field
UWB	S	14.10	5.88	8.88	12.17	5.11
	M	8.54	5.40	11.36	8.21	6.84
Dyn.	S	**11.92**	7.16	**8.16**	**10.89**	10.18
	M	**6.88**	6.32	**8.24**	**6.69**	16.27
Stat.	S	14.10	**5.65**	11.74	12.83	9.49
	M	11.57	11.31	21.9	13.11	13.14
All	S	**11.77**	6.75	**7.4**	**10.91**	9.07
	M	**6.96**	7.06	11.96	**7.75**	13.10

Table 6 UWB errors (cm) on different test-datasets (columns) corrected with the error mitigation model trained on various train-datasets (rows). Top row shows the uncorrected but already calibrated statistical values for each test dataset. Improvements are highlighted in bold. (S = Std., M = MAE)

3.3.2 Inference Benchmark

Most mobile robots or systems are endowed with computing units utilizing Linux-capable Arm Cortex-A CPUs. A widely used computing unit for this purpose is the Raspberry Pi. To determine the performance of the network models on such a unit, an inference benchmark was performed. The benchmark took place on a Raspberry Pi 4 with a Cortex-A72 (ARM v8), 4GB RAM and Ubuntu 20.04.4 LTS (64 bit). The models were deployed using the TensorFlow-Lite[1] 2.5.0 framework with Python 3.9. The model was executed for 1000 iterations in a row. To measure the execution time of only the model itself, a static set of dummy input parameters was used, and the model's output was fetched and written to a variable. As expected, both model's performance is very similar. An iteration with the classification network took on average 0.2290 ms, for the regression network it took 0.2295 ms. This results in a maximum execution performance of 4,367.57 respectively 4,357.22 inferences per second. This should leave enough computational headroom for further processing of the networks output as well as for other tasks. By reducing the model size, also the use on microcontrollers is conceivable. To show potential feasibility two dummy models were created, consisting of only five fully connected layers with 15,106 (trainable) parameters each. Both models were consecutively executed on an ESP32 (240MHz, 320KB RAM, 4MB Flash) microcontroller for multiple times, whereby it took 1.43 ms to execute both models on average.

[1] https://www.tensorflow.org/lite

4 Conclusion and future outlook

The presented experimental results suggest that UWB ranging errors can be mitigated by using the approach introduced in this paper. At the moment, however, the created network models seem to be environmental dependent and not universally applicable. Further experiments, which could not be included in this paper due to time constraints, such as other work [9] shows that models can be capable of universal generalization at least for indoor scenarios. By extension of the data in the training-set, tweaking in the model's architecture such as the training process (e.g. by utilizing data augmentation), further potential may be exploited. In particular, the combined use of both the on-chip metrics such as CIR values is, according to our current knowledge, used only in the proposed approach so far.

5 Literature

[1] Decawave Ltd: DW1000 User Manual, 2017, `https://www.decawave.com/sites/default/files/resources/dw1000_user_manual_2.11.pdf`

[2] Stampa, M.; Müller, M.; Heß, D.; Röhrig, C.: Semi-automatic Calibration of UWB Range Measurements for an Autonomous Mobile Robot. Proceedings of the 50th International Symposium on Robotics (ISR 2018), 2018, S. 300-305, Munich, Germany, ISBN: 978-3-8007-4699-6

[3] Decarli, N.; Dardari, D.; Gezici S.; D'Amico, A. A.: LOS/NLOS detection for UWB signals: A comparative study using experimental data. IEEE 5th International Symposium on Wireless Pervasive Computing 2010, 2010, pp. 169-173, doi: 10.1109/ISWPC.2010.5483704.

[4] Sang, CL.; Steinhagen B.; Homburg, JD.; Adams, M.; Hesse, M.; Rückert, U.: Identification of NLOS and Multi-Path Conditions in UWB Localization Using Machine Learning Methods. Applied Sciences. 2020; 10(11):3980. doi 10.3390/app10113980

[5] L. Schmid, D.; Salido-Monzú; Wieser, A.: Accuracy Assessment and Learned Error Mitigation of UWB ToF Ranging. International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2019, pp. 1-8, doi: 10.1109/IPIN.2019.8911769.

[6] Zeng, Z.; Liu, S.; Wang, L.: NLOS Identification for UWB Based on Channel Impulse Response. 12th International Conference on Signal Processing and Communication Systems (ICSPCS), 2018, pp. 1-6, doi: 10.1109/ICSPCS.2018.8631718.

[7] Bregar, K. and Mohorcic, M.: Improving indoor localization using convolutional neural networks on computationally restricted devices," IEEE Access, vol. 6, pp. 17 429–17 441, 2018.

[8] Ridolfi, M.; Fontaine, J.; Herbruggen, B.V.; et al: UWB anchor nodes self-calibration in NLOS conditions: a machine learning and adaptive PHY error correction approach. Wireless Networks 27, 2021, pp. 3007–3023. https://doi.org/10.1007/s11276-021-02631-0

[9] Angarano, S.; Mazzia, V.; Salvetti, F.; Fantin, G.; Chiaberge, M.: Robust Ultra-wideband Range Error Mitigation with Deep Learning at the Edge. Eng. Appl. Artif. Intell., 2021, 102, 104278.

[10] Maranò, S.; Gifford, W.M.; Wymeersch, H.; Win, M.Z.: NLOS identification and mitigation for localization based on UWB experimental data. IEEE J. Sel. Areas Commun., 2010, 28, 1026–1035.

[11] Wymeersch, H.; Maranò, S; Gifford, W.M.; Win, M.Z.: A Machine Learning Approach to Ranging Error Mitigation for UWB Localization. IEEE Transactions on Communications, 2012, 60, 1719–1728.

[12] Kingma, D. P.; Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

