Enhancing ToA positioning with a hybrid deep learning approach

Dolores Garcia †⋆, Alejandro Blanco †⋆, Héctor Cordobés† Joerg Widmer†

† IMDEA Networks Institute, Madrid, Spain

⋆ Universidad Carlos III de Madrid, Madrid, Spain

E-mails: †{firstname.lastname}@imdea.org

***Abstract*—We present a positioning algorithm that achieves 7 cm of mean positioning error and better than 3 cm in 80% for randomly chosen validation data. We achieve such accuracy combining two different algorithms into a hybrid approach which exploits the strengths of both. The first algorithm exploits the time of arrival information per antenna to subsequently feed it to a classical fingerprinting algorithm. The second one is a convolutional neuronal network which exhibits better stability in long-tail samples. These two methods are merged into a hybrid approach leveraging the divergence between the two predictions.**

***Index Terms*—Time of Arrival, MUSIC, fingerprinting, positioning, CNN, machine learning, GBM**

I. Visualizing the Data

The aim of the CTW competition is to localize a robot given channel measurements from an antenna array. The data contains 17486 measurements consisting of:

- A channel response matrix with size: number of antennas times number of used subcarriers (16×924)
- A vector of signal-to-noise-ratios of all the antenna elements
- Position coordinates of the robot (X, Y, Z) as ground truth

Our first approach was to characterize a 2D model where the unknown variables were the distance and the direction. To do so, we estimated the Angle of Arrival (AoA) and the Time of Arrival (ToA) of the signal using the classical MUlti SIgnal Classification (MUSIC) algorithm. The AoA estimation presents a great variability in the value for a specific location limiting the positioning accuracy when solely relying on this information. On the other hand, the ToA estimation is stable, thus we choose only this metric as the input for our algorithm.

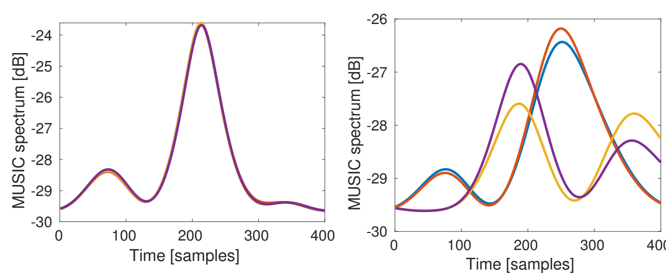
II. Preprocessing the Data

A. Reducing the dimensionality of the data

The ToA estimation itself does not give spatial information since we can only model 1D space with the distance. To overcome this, we estimate the ToA individually per antenna element. Depending on the AoA of the incoming signal, this produces a pattern in the individual ToA values. In other words, the closest antennas to the robot will have lower ToA values and the furthest ones higher values which will characterize a position. With this approach, we reduce the problem to that of associating a given ToA pattern of 16 values with a position coordinate, reducing the dimension of the data from 924x16 complex channel values to 16 ToA real values per measurement.

B. Shifted ToA

Careful observations revealed that several ToA estimations were consistently delayed without following the behaviour of the rest of the antennas. We believe that these could be linked to delays added by the hardware processing or signal routing and should be compensated. Fig. 1 shows an example of such behavior where the MUSIC temporal spectrum is plotted for several measurements in one specific location to represent the cases whether the ToA of some measurements are delayed or not. The spectra in Fig. 1a does not have any delay and we consider it as the regular behavior, while the spectra of some of the measurements in Fig. 1b are delayed. We remove such effect applying a temporal circular shift to the MUSIC spectrum to center the ToA estimation in the correct moment in time once we determine the delay.



(a) Not delayed

(b) Delayed

Fig. 1: MUSIC temporal spectra of an antenna for four measurements in the same location. (a) illustrates the case without delayed ToA and (b) shows a case in which two measurements are delayed

III. Implementation

We compare two methods to determine the location, a fingerprinting and Convolutional Neural Network (CNN). In addition, we describe a hybrid approach which exploits the strengths of both methods.

Layer	Size	Filters
Conv1D	9	16
MaxPool1D	2	
Conv1D	9	32
MaxPool1D	2	
Conv1D	7	64
MaxPool1D	2	
Dense	1024	
Dense	1024	
Output	3	

TABLE I: Neural network architecture

A. *Fingerprinting*

Considering this many to one mapping per position, the first method treats the preprocessed data using a classical fingerprinting method. This is done by comparing a given test measurement $(t_1^j, \ldots, t_{16}^j)$ with all training measurements $(T_1^i, \ldots, T_{16}^i)$ and assigning to a test measurement the position of the measurement that minimizes the distance of the ToA of all antenna elements, as:

$$r_j = r_I \quad I = \underset{\forall i}{\operatorname{argmin}} \sum_{a=1}^{16} \left| T_a^i - t_a^j \right|,$$

where $r_i = (x_i, y_i, z_i)$ is the position of the ith measurement.

B. *Convolutional Neural Network*

In parallel to the ToA approach, we also tried a raw-data-based approach. We realized that the baseline CNN proposed by the organization lacked expressive capacity and found that by using several convolutional layers at the input of the network, trying to emulate a phase processing like in ToA, we obtain good results (see table I). In our case we limited the convolutional filtering to one dimension (along the different carriers) and the data to the raw samples, using real and imaginary values separately.

C. *Hybrid approach*

Once we had both systems trained and tuned, it was noticeable that the models excelled in different conditions (see Fig. 2). That suggested that a hybrid model integrating both results could improve the overall result. We also observed that the dependence of the accuracy with regard to the divergence between both estimations was different for both approaches. We can leverage this fact to create a classifier based on the divergence of the predictions. When the distance among predicted positions is higher than a given threshold, one of the models will be chosen. The fingerprinting model showed larger errors when the divergence increased, so fingerprinting is used for smaller divergences and CNN for larger ones. The optimal threshold (w.r.t. RMSE) is computed over each validated partition, and the overall performance is then calculated over the whole data with one single threshold, which is the mean of all the thresholds previously computed. This value is around 0.6 m and has a very low standard deviation (less than 8 cm).

Model	RMSE	MAE	80-percentile
Fingerprinting	0.66	0.20	**0.02**
CNN	0.24	0.16	0.22
Hybrid (threshold)	0.21	**0.07**	**0.03**
Hybrid (GBM)	**0.19**	0.09	0.11

TABLE II: Representative error values in meters for all algorithms. Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and the 80-percentile of the error distribution

We also evaluated a Machine Learning approach, where we trained a regressor with the estimations from both approaches. The best RMSE result was obtained with three Gradient Boosting Machine (GBM) models (one per dimension).

IV. Numerical Results

To evaluate our results (shown in table II), 10-fold cross-validation has been used, randomly chosen from the initial set. We would like to point out the best RMSE value (0.19 m) in the Machine Learning approach of the GBM Hybrid Model, but also (depending on the application) the good results from the Threshold-based Hybrid Model or the 80-percentile result of the Fingerprinting method (just 2 cm). In fact, the Threshold-based approach is very interesting as it can be more robust (just one parameter to adjust) and faster to train than the GBM approach (for instance, on dynamic scenarios).

For Fingerprinting, we observed measurement errors of meters that are randomly distributed over the table and intensify at its edges. Our observations help us identify these large errors as positions in which many measurements are performed in one location and data shows more than two clusters of different ToA values corresponding to the same spatial location.

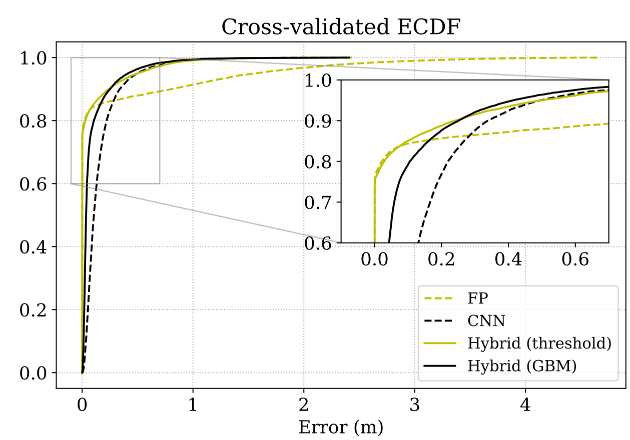


Fig. 2: Cumulative distribution function of the error for all positioning methods, error in log scale

As a final note, both the preprocessing and the positioning algorithms are methods which do not exploit any spatiotemporal information of the trajectory of the robot. Better accuracy would by leveraging this information (for example, by better detecting outlier estimations, using Kalman Filters, etc).