Grid-free Harmonic Retrieval and Model Order Selection using Convolutional Neural Networks

Steffen Schieler\textsuperscript{1}, Sebastian Semper\textsuperscript{1}, Reza Faramarzhangari\textsuperscript{1}, Michael Döbereiner\textsuperscript{2}, Christian Schneider\textsuperscript{2}, and Reiner Thomä\textsuperscript{1}

\textsuperscript{1}Technische Universität Ilmenau, FG EMS
\textsuperscript{2}Fraunhofer Institute of Integrated Circuits: Dep. EMS

January 26, 2024

Abstract

Harmonic retrieval techniques are the foundation of radio channel sounding, estimation and modeling. This paper introduces a Deep Learning approach for joint delay-and Doppler estimation from frequency and time samples of a radio channel transfer function. Our work estimates the two-dimensional parameters from a signal containing an unknown number of paths. Compared to existing deep learning-based methods, the signal parameters are not estimated via classification but in a quasi-grid-free manner. This alleviates the bias, spectral leakage, and ghost targets that grid-based approaches produce. The proposed architecture also reliably estimates the number of paths in the measurement. Hence, it jointly solves the model order selection and parameter estimation task. Additionally, we propose a multi-channel windowing of the data to increase the estimator’s robustness. We also compare the performance to other harmonic retrieval methods and integrate it into an existing maximum likelihood estimator for efficient initialization of a gradient-based iteration.
Grid-free Harmonic Retrieval and Model Order Selection using Convolutional Neural Networks

S. Schieler∗, S. Semper†, R. Faramarzhangari∗, M. Döbereiner†, C. Schneider∗†, R. Thomä∗
∗Technische Universität Ilmenau: FG EMS, Ilmenau, Germany, steffen.schieler@tu-ilmenau.de
†Fraunhofer Institute of Integrated Circuits: Dep. EMS, Ilmenau, Germany

Abstract—Harmonic retrieval techniques are the foundation of radio channel sounding, estimation and modeling. This paper introduces a Deep Learning approach for joint delay- and Doppler estimation from frequency and time samples of a radio channel transfer function.

Our work estimates the two-dimensional parameters from a signal containing an unknown number of paths. Compared to existing deep learning-based methods, the signal parameters are not estimated via classification but in a quasi-grid-free manner. This alleviates the bias, spectral leakage, and ghost targets that grid-based approaches produce. The proposed architecture also reliably estimates the number of paths in the measurement. Hence, it jointly solves the model order selection and parameter estimation task. Additionally, we propose a multi-channel windowing of the data to increase the estimator’s robustness.

We also compare the performance to other harmonic retrieval methods and integrate it into an existing maximum likelihood estimator for efficient initialization of a gradient-based iteration.

Index Terms—Parameter Estimation, Convolutional Neural Networks, Delay-Doppler Estimation, Harmonic Retrieval.

I. INTRODUCTION

Harmonic Retrieval is a problem encountered in many signal processing tasks, e.g., channel estimation [1], radar localization, and direction finding. Available solutions for the task can be divided into four groups: subspace algorithms, like Multiple Signal Classification (MUSIC) [2] or Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) [3], iterative maximum likelihood (ML) [4], and Sparse Signal Recovery (SSR) [5] and finally most recently Deep Neural Network (DNN)-based algorithms [6]–[11]. Our work belongs to the latter category and presents a new approach to solve a Harmonic Retrieval task by using a deep Convolutional Neural Network (CNN). We note, the applications in the listed works, namely spectral- and Direction of Arrival (DoA)-estimation, share the algebraic structure of Harmonic Retrieval.

In [6] a CNN is used to estimate frequencies components by predicting a super-resolution pseudo-spectrum from a superposition of up to ten components. A separate, subsequent network performs the model order estimation task. The results show a performance improvement when compared to MUSIC, especially in the low-Signal- to-Noise Ratio (SNR) domain. In [7] a CNN is trained to perform DoA estimation of up to three unknown sources by determining their location on a grid, i.e., solving a classification problem. Similarly, the authors of [8] address the problem of DoA estimation by combining a denoising autoencoder with another DNN for the estimation. As in [6], both approaches show performance improvements in the low-SNR domain compared to MUSIC. However, classification methods suffer from an inherent estimation bias due to grid-mismatch and poor scaling in terms of desired resolution. Since increasing the grid density increases classification complexity non-linearly. Naturally, estimates that do not rely on a grid, i.e., classification, are highly interesting, especially if we wish to achieve super-resolution. Comparably, the work in [11] combines a regression task solved by a DNN with gradient steps on the likelihood function of the data to perform DoA estimation. The grid-free estimates from the DNN are used as initial guesses for a second-order Newton method. This approach combines the best of two worlds: fast, robust, but approximate initial estimates combined with an iterative high-resolution method allowing quadratic convergence, but only after successful initialization provided by the DNN. The results highlight the decrease in computational complexity and improvements in performance from the use maximum-likelihood methods. However, the presented method only considers up to 3 stochastic sources estimated from multiple snapshots, which is insufficient in many practical scenarios, especially if parameters are modeled as deterministic. More significantly, the used antenna array geometries have a very small aperture, rendering the initialization of the Newton method very well conditioned since it tolerates high deviations of the initialization from the true solution.

Compared to previous works, our proposed architecture can estimate up to 20 deterministic paths from a single snapshot in terms of their delay and Doppler-shift from frequency-time data. The number of paths is estimated together with their propagation parameters. We exploit the sparsity of the parameter space by dividing it up into a low number of auxiliary grid-cells. Then we employ a CNN to estimate the number of paths in each cell and their respective deviations from the cells’ centers to obtain grid-free estimates. In effect, we solve multiple joint regression and classification tasks to obtain the number of sources and their parameters. To render the training more robust, we apply a set of windowing functions both in frequency and time domain to the input data, effectively presenting the same data with different pulse-shapes. To verify the performance, we consider the mean squared error (MSE) of the raw DNN estimates and compare them to existing methods, i.e., Discrete Fourier Transform (DFT) and RIMAX [4], and the Cramér-Rao Bound (CRB) as the theoretical lower bound. We also conduct an isolated analysis of the model order selection performance and compare our proposal to RIMAX and Efficient Detection Criterion (EDC). Motivated by the approach in [11], we also use the estimates directly to warm-start a second-order gradient iteration and show that the initial guesses allow convergence with high probability. The results indicate that the proposed methodology can reliably predict both model order and parameters sufficiently well and with comparably short computation time. When enhanced with a few steps of an iterative ML estimator, the performance can be improved substantially, resulting in a well-performing, robust 2D parameter estimator with moderate computational complexity.

II. SIGNAL MODEL

Our task is retrieving the spectral paths of deterministic paths from frequency and time samples of a radio channel, i.e., their propagation delays $\tau$ and Doppler-shifts $\alpha$.

We model the wireless channel transfer-function measurement of bandwidth $B$ with $N_f \in \mathbb{N}$ frequency samples and $N_t \in \mathbb{N}$ snapshots and employ the narrowband assumption since $B \ll f_c$. We denote the sampled observation in complex baseband $(f_c = 0)$ by $S$. 

\begin{equation}
S = \sum_{k=1}^{N_t} \sum_{i=1}^{N_f} s(k \Delta t - f_k) e^{j 2 \pi \alpha f_k t_i} + n(k \Delta t - f_k), \quad k = 1, \ldots, N_t, \quad i = 1, \ldots, N_f.
\end{equation}
The motivation for the multi-window approach is to obtain different information from the same samples. Rectangular windows achieve the maximum SNR and provide a narrow pulse shape but also result in high sidelobes after the DFT, potentially introducing ghost paths. Other filters, such as Hann-windows, reduce the sidelobes, i.e., the probability of ghost paths, but increase the mainlobe width and usually resulting in higher estimation variances. Hence, the choice for a window function is usually application specific and bears trade-offs. However, a CNN can process multiple views of the same data in parallel, similar to color channels of an image. With this approach, diverse views can be exploited for more robust estimates. We used $N_W = 8$ different windowing functions, i.e., a Tukey, Taylor, Chebyshev, Blackman, Flat Top, Cosine, Hann, and the Rectangular window.

The second step is to apply a 2D-DFT over the last two dimensions, transforming it to the target parameter domain in delay- and Doppler, denoted as $\mathbf{Y}_1 \in \mathbb{C}^{N_{f} \times N_t}$. To obtain real-valued numbers for the training, we employ four mapping functions $f_1$, $f_2$, $f_3$, and $f_4$ to map the complex values in $\mathbf{Y}_1$ to the real-valued CNN input data $\mathbf{Y}_2 \in \mathbb{R}^{4 \times N_{f} \times N_t}$. Here, $R$ and $\Im$ denote the real and imaginary parts, respectively, and $| \cdot |$ and $\angle$ denote the absolute value and phase of a complex number. Even though $f_1$ and $f_2$ contain the same information as $f_3$ and $f_4$, our experiments showed, the simultaneous usage adds useful diversity for the CNN.
B. Off-Grid Parameter Encoding

The labels for the supervised learning task are provided by a grid-relative parameter encoding. To this end, we divide the delay-Doppler domain into a few grid-cells. In each grid-cell, we aim to estimate the number of paths in that cell and their respective deviations from the cell center. Hence, the parameter values are encoded relative to the grid cell centers. Our encoding is a task-specific modification of [12]. It consists of three steps: parameter normalization, cell assignment, and relative parameter encoding.

Let $C$ be the maximum number of paths in a single cell. The normalization maps the parameters into a range between 0 and 1, such that $\tau$ and $\alpha$ are in the interval of $\tau_0 \in [0, 1]$ and $\alpha_0 \in [0, 1]$. Next, we define a set of $I \times J$ cell centers $x \in (0, 1)^I \times J$, which each define a non-overlapping rectangular covering of $[0, 1) \times [0, 1)$. Paths are mapped to the cells based on the shortest $\ell_2$-distance to the cell centroids $x_{i,j}$. To encode the paths' positions and number in each cell, we define vectors $\eta_{i,j} \in \mathbb{R}^{I \times J \times C}$ as

$$\eta_{i,j} = \left[ \mu_{C}^{[i,j]}, \Delta \tau^{[i,j]}_1, \Delta \alpha^{[i,j]}_1, \ldots, \mu_{C}^{[i,j]}, \Delta \tau^{[i,j]}_C, \Delta \alpha^{[i,j]}_C \right]^T$$

where $\|x - x_{i,j}\|$ and $\|x - x_{i,j}\|_2$ denote the Euclidean distance between the cell centroid $\eta_{i,j}$ and the respective parameters. In line with best practices for training, $\Delta \tau$ and $\Delta \alpha$ are normalized by the cell width and shifted based on the cells' centers, such that $\Delta \tau_{i,j}, \Delta \alpha_{i,j} \in [0, 1]$. Note, that the model order $P$ can be expressed as $P = \sum_{i,j \in C} \mu_{C}^{[i,j]}$ and the maximum number of encodable paths is $C \cdot I \cdot J$.

As the number of paths in each cell can vary, $\mu_{C}^{[i,j]} = \{0, 1\}$ indicates if the path with encoding $\Delta \tau^{[i,j]}_1$ and $\Delta \alpha^{[i,j]}_1$ is an estimate ($\mu_{C}^{[i,j]} = 1$) or empty ($\mu_{C}^{[i,j]} = 0$). This enables computing a coupled loss during training, as detailed in Section III-D, for arbitrary cell assignments. To enforce a predictable ordering of paths in each cell, the paths are sorted from $c = 1 \ldots C$ with descending magnitude $\gamma_p$, and unassigned parameters are labeled as 0 (see Figure 2). The result of the encoding is a 3D array $\eta \in \mathbb{R}^{I \times J \times C}$, which structures the desired prediction results of our CNN.

C. Network Architecture

Our network architecture is split into four stages represented in Figure 1. The first stage passes the input through 5 blocks of 2D convolutional layers. Each block consists of a 2D convolutional layer, followed by Batch-Normalization and a Rectified Linear Unit (ReLU) activation function. The convolutional layers are parameterized to preserve the data shape but double the number of channels after each block.

The second stage performs downsampling via convolutional layers with a stride of 2, which reduces the data dimension by 2 with each block. In this stage, the number of channels is preserved.

Stage three achieves the parameter predictions. First, the number of channels is reduced by two blocks of convolutional layers, followed by two fully-connected (FC) layers interleaved by a ReLU activation function. The relative parameter estimates $\eta$ (from Section III-B) are encoded in the output of the final FC layer.

The fourth stage is used to predict the number of paths $P$ (model order) based on the results of the second stage. It uses a single convolutional block followed by two FC layers interleaved by a ReLU activation function. Its output is $\hat{P}$, a one-hot encoded vector of the model order estimate $\hat{P}$.

D. Loss Functions and Training

Our approach uses multiple loss functions combined in a weighted sum. The first summand is the loss for the model order estimate $\hat{P}$. It uses the well-known Binary Crossentropy (BCE) loss for the one-hot encoded values.

$$L_0 = \hat{P} \cdot \log(\rho) + (1 - \hat{P}) \cdot \log(1 - \rho)$$

The loss for the parameter estimates $\eta$ utilizes a masked MSE loss function

$$L_1 = \sum_{i,j} \sum_{c=1}^{C} \left( \sigma \left( \frac{\mu_{C}^{[i,j]}}{\Delta \tau^{[i,j]}_c} \right) \cdot \left( \left\| \Delta \tau^{[i,j]}_c - \Delta \tau^{[i,j]}_{c, \eta} \right\|_2 \right) \right)^2,$$

where $\sigma(\cdot)$ represents the sigmoid function, and $\cdot$ marks the predictions. As mentioned earlier, $\mu_{C}^{[i,j]}$ is used to weight the parameter estimates $\Delta \tau^{[i,j]}_c$ and $\Delta \alpha^{[i,j]}_c$ during loss calculation. Its effect becomes apparent by inspecting the limits of the $\sigma(\cdot)$ function, as

1) $\lim_{x \to -\infty} \sigma(x) = 1$, causing the MSE of the corresponding predictions to contribute to $L_1$.
2) $\lim_{x \to +\infty} \sigma(x) = 0$, causing the MSE of the corresponding predictions to not contribute to $L_1$.

Hence, predicting a negative value for $\mu_{C}^{[i,j]}$ causes $\sigma(\mu_{C}^{[i,j]})\Delta \tau^{[i,j]}_c \approx 0$ and $\sigma(\mu_{C}^{[i,j]})\Delta \alpha^{[i,j]}_c \approx 0$ and hence close to the corresponding 0 in the labels.

Finally, both loss components are combined in a weighted sum via

$$L_{total} = L_0 + \beta \cdot L_1.$$ 

For our experiments, we manually selected $\beta = 4$ to ensure both losses equally contribute to the learning. We note, that $\beta$ is likely not optimal and choosing it constitutes a multi-objective optimization problem for further study.

E. Training

The three synthetic datasets, a training, validation, and test set, were created by sampling random values for the signal parameters. Table I contains a comprehensive summary of the respective settings for the dataset and training hyperparameters.

Each sample in the dataset contains a random number of 1 to 20 specular paths. The complex path amplitudes $\gamma$ contain random phases, and their magnitudes are uniformly spread across the range of $0 \text{ dB}$ to $30 \text{ dB}$. To prevent overfitting, the measurement noise $N$ is generated randomly for every snapshot with a random noise variance $\sigma$, such that the SNR is in the range of $10 \text{ dB}$ to $50 \text{ dB}$. We use a uniform distribution in the linear domain, such that 90% of samples have a SNR $<10 \text{ dB}$.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution $\tau_p$, $\alpha_p$</td>
<td>$\mathcal{U}(0,1)$</td>
</tr>
<tr>
<td>Min. separation $\tau_p$, $\alpha_p$</td>
<td>$3.125 \times 10^{-3}$</td>
</tr>
<tr>
<td>Magnitudes $\mathcal{U}(0, 0.01, 1]$</td>
<td></td>
</tr>
<tr>
<td>Phases</td>
<td>$\mathcal{U}(0, 2\pi)$</td>
</tr>
<tr>
<td>SNR</td>
<td>0 dB to 50 dB</td>
</tr>
<tr>
<td>Number of Paths</td>
<td>$\mathcal{U}(0, 20)$</td>
</tr>
<tr>
<td>Trainingset Size</td>
<td>$400 \times 10^3$</td>
</tr>
<tr>
<td>Validationset Size</td>
<td>1000</td>
</tr>
<tr>
<td>Testset Size</td>
<td>4000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Training</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimizer</td>
<td>Adam [13], $\gamma = 0.0003$, $\beta_1 = 0.9$, $\beta_2 = 0.999$</td>
</tr>
<tr>
<td>Mini-Batchsize</td>
<td>32</td>
</tr>
<tr>
<td>Epochs</td>
<td>20</td>
</tr>
<tr>
<td>Trainable Parameters</td>
<td>$25 \times 10^6$ for $N_f = N_t = 64$</td>
</tr>
</tbody>
</table>
IV. Analysis

In order to get a complete estimate of the parameters in (3), we retrieved estimates $\hat{P}$, $\hat{\tau}$ and $\hat{\alpha}$ and based on these used the Best Linear Unbiased Estimator (BLUE), i.e., least-squares, to attain an estimate for the linear weights $\hat{\gamma}$.

To illustrate the results obtained from our CNN, Figure 3 shows a single, hand-picked sample from the validation set passes through the network at different SNRs. Not only do we analyze the raw parameter estimates of the network, but we also use these estimates to warm-start a gradient iteration

$$\left(\gamma^{k+1}, \tau^{k+1}, \alpha^{k+1}\right) = \left(\gamma^k, \tau^k, \alpha^k\right) - \varepsilon^k z^k \tag{8}$$

with descent direction

$$z^k = \left[\left(F^k\right)^{-1} \cdot J^k\right] \left(\gamma^k, \tau^k, \alpha^k\right),$$

which essentially defines a second-order Gauss-Newton scheme, since $F$ is the Fisher-Information matrix and $J$ is the Jacobian matrix of the negative log-likelihood function based on the assumption that $N$ is a Gaussian random variable, which reads as

$$\lambda(\gamma, \tau, \alpha) = \frac{1}{\sigma^2} \|Y - S(\gamma, \tau, \alpha)\|^2_{P^\prime}.$$

To provide an assessment of the estimation performance of our approach, we compare it to a periodogram-based peak search, i.e., DFT, which is inherently grid-limited, and the high-resolution RIMAX [4] based on ML. The model order required for the peak search is obtained from the EDC [14]. We provide a comparison in terms of the MSE for the estimated parameters and the respective model order error.

Figure 4 shows that our approach can overcome the grid limitation and outperforms the periodogram-based method. However, the raw estimates' accuracy of our approach also saturates, albeit at a lower MSE. As expected, the high-resolution estimator's results align with the predictions of the CRB with increasing SNR.1 When using the estimates provided by our approach to initialize (8), the estimation accuracy for higher SNRs improves significantly from only a 10 iterations of (8). This is a highly promising feature of our architecture since the joint estimates are close enough to the global minimum of (9) such that (8) is very likely to converge to the true solution. This starkly contrasts algorithms like RIMAX, which only add single sources one by one to the set of estimates and carry out iterative refinement between two newly added sources, known as successive interference cancellation. In our case, we can initialize the iteration much more efficiently with a single forward from the network but still accurate enough for the gradient iteration to converge.

Apart from the accuracy, the computational complexity of the algorithms is also of interest. As an assessment of the computational complexity of the RIMAX estimator is not straightforward due to the use of iterative numerical methods, we assess the runtime per sample on an identical system. The periodogram approach ranks fastest with an average of 3 ms, followed by our approach with 19 ms. When combining our method with 10 iterations of (8) we average at 60 ms, while the ML algorithm RIMAX requires around 11.9 s on average. It highlights that our approach addresses applications requiring fast fixed-clock estimates, where the accuracy-runtime trade-off can be regulated by the number of gradient iterations.

Regarding the model order estimation, we compare our approach to EDC and the model order extracted from the ML estimates. The results are illustrated in Figure 5 and reaffirm the findings of previous publications [6], [11], where it is shown that neural networks can reliably predict the number of sources in a signal. Our approach consistently achieves the best results across the studied SNR range. EDC and RIMAX achieves similar performance at SNRs $>20$ dB, but underestimates the model order for smaller SNR. Overall, this result highlights the model order estimation capabilities of our approach, particularly in the challenging low-SNR domain.

V. Conclusion

Our work introduces a new approach for combined, two-dimensional harmonic retrieval and model order estimation using a CNN. Compared to recent approaches in the field, it uses a cell-based representation of spectral parameters for the prediction and, therefore, can estimate parameters directly via regression instead of classification. Regarding estimation accuracy, it outperforms on-grid periodogram-based approaches. Most interestingly, the estimates of up to 20 paths can be used to warm-start a second-order gradient iteration of the highly non-convex likelihood function. With some further refinement, the architecture can initialize a well-performing ML, hence approximately delivering all the advantageous properties, like statistical consistency and efficiency. Especially compared to [11], this is an improvement in terms of parameter dimension, quantity, and data complexity.
Additionally and in line with previous work, it demonstrates superior model order estimation, especially in the low-SNR regime. Our approach is well-suited for time-constraint harmonic retrieval tasks because of its relatively low runtime compared to high-resolution methods.

Due to the CNN structure of our approach, it scales well to more than two dimensions. Ultimately, this should allow the CNN processing of multiple input multiple output (MIMO) measurements and hence full channel sounding data, including spatial measurements, similar to [4]. Then, the preprocessing must be extended with realistic antenna beampatterns by a suitable beamspace transformation such as the Effective Aperture Distribution Function (EADF), see [15]. Moreover, an ablation study should quantify the performance impacts of the individual architecture blocks. Similarly, processing real measurement data will help us understand to what degree the approach is affected by measurement system imperfections and model mismatch. Further opportunities are the applicability to wideband channel data, where the narrowband assumption is no longer satisfied, leading to dispersion in delay and Doppler domains and coupling of the two parameters.

**ACKNOWLEDGMENT**

The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany in the project “Open6GHub” (grant number: 16KISK015), “KOMSENS-6G” (grant number: 16KISK125), and DFG project HoPaDyn with Grant-No. TH 494/30-1.

**REFERENCES**


