DOA Estimation Using an Unfolded Deep Network in the Presence of Array Imperfections

Liuli Wu*
China Electronic Device System-Engineering Corporation
Beijing, China
$wll\_me1991@163.com$

ZhangMeng Liu
College of Electronic Science, National University of Defense Technology
Changsha, China

Jianhua Liao
China Electronic Device System-Engineering Corporation
Beijing, China

2022 7th International Conference on Signal and Image Processing (ICSIP) | 978-1-6654-9563-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICSIP55141.2022.9886344

***Abstract*—Deep learning has gained remarkable popularity due to its widespread success in a number of practical applications such as image classification and speech recognition. In this paper, we consider the application of deep learning to DOA estimation problem in the presence of array imperfections. Firstly, through spatially overcomplete formulation, the DOA estimation problem is converted to a sparse linear inverse problem, where one seeks to recover a sparse signal from a few noisy linear measurements. Then an iterative sparse signal recovery algorithm, iterative shrinkage/thresholding algorithm(ISTA), is ”unfolded” to form an interpretable and learnable deep network–learned ISTA(LISTA). LISTA is applied to recover the DOA spectrum after trained with vast amounts of training data. We find that LISTA is of good robustness against noise and array imperfections due to the training procedure. Comprehensive simulations and experiments have been carried out, and the superiority of the proposed method can be clearly seen.**

***Index Terms*—direction-of-arrival estimation, iterative shrinkage/thresholding algorithm, sparse recovery, unfolded network**

I. Introduction

Passive direction-of-arrival (DOA) estimation has received enormous attention over the last decades. Myriad methods have been developed to solve this problem, such as beamforming methods [1], subspace-based methods [2], maximum likelihood methods [3], sparsity-inducing methods [4], [5] and machine learning based methods [6]–[13].

Generally, the above mentioned methods fall into two categories. Vast majority of them belong to the ”model-driven” category in the sense that they establish parametric models for the signal or the array structure [1]–[5]. Usually, a forward mapping from signal directions to array outputs is formulated and DOA estimation is accomplished by matching the array outputs with the pre-formulated model based on the expert criteria. The appeal of ”model-driven” methods is that they are based on interpretable expert knowledge and often have well understood behavior. However, these methods also suffer from the drawbacks of heavy computational burden due to the time-consuming operations such as matrix inversion [2], multi-dimensional peaking searching [3] or large number of iterations [5], [14] and performance degradation when array imperfections such as the amplitude and phase inconsistencies or mutual coupling among the arrays elements exist.

The other methods can be considered as the ”data-driven” category (often machine learning based) in the sense that they learn knowledge from the training data [8]–[12], [15]–[17]. Instead of using pre-assumed models, ”data-driven” methods utilize machine learning techniques to learn the nonlinear relationship between array outputs and signal directions through vast amounts of training data. Once trained, the network can be used to predict source directions correspond to newly obtained array outputs. The appeal of ”data-driven” DOA estimation methods is that they learn the array structure from training data directly and make no pre-assumptions on the forward mappings. Therefore, they are expected to have built-in adaptations to array imperfections. Moreover, the machine learning based DOA estimation process is just the feedforward process of the network, making it much faster and possible to achieve real time estimation. However, these methods are held back by the fact that there exists almost no theory governing their performance (like unprincipled black boxes) and that they are highly sensitive to the training data.

Recently, researchers in the field of compressive sensing try to blend ”model-driven” algorithms with ”data-driven” algorithms to solve sparse linear inverse problems where one seeks to recover a sparse signal from a few noisy linear measurements [18]–[21]. These methods first unfold a well-understood iterative recovery algorithm such as ISTA, AMP, VAMP or the SBL to obtain a signal-flow graph (or network) with trainable variables, then the trainable variables can be tuned with a supervised learning method such as stochastic gradient descent algorithms based on back propagation. Gregor and LeCun presented the Learned ISTA with learnable threshold variables for a shrinkage function [18]. Borgerding et al. also presented variants of AMP and VAMP with learnable capability [19]. Such methods benefit from the ability to learn more realistic signal priors from the training data, while still maintaining the interpretability of the ”model-driven” methods. Moreover, since the unfolded forward network is explicitly accounted for, a smaller or shallower network with fewer parameters is sufficient to capture the signal information compared to their completely ”data-driven” counterparts, thus providing benefits including lower demand for training data, reduced risk of overfitting, and implementations with significantly reduced memory requirement. Previous works (e.g. [18]–[21]

978-1-6654-9563-9/22/$31.00 ©2022 IEEE

) have shown that these approaches have the potential to offer significant improvements, in both accuracy and complexity, over traditional recovery algorithms.

DOA estimation is a typical sparse linear inverse problem after appropriate formulation and moderate approximation, as illustrated in the sparsity-inducing DOA estimation methods [4], [5], [14], [22]. Taking inspirations from the above works [18]–[21], we propose an unfolded deep network based method to deal with DOA estimation problem in the presence of array imperfections. This approach is obtained by unfolding the ISTA [23] algorithm into a deep network and learning the network parameters that best fit a large training dataset. Through training, the proposed method learns the inverse transformation from measurements to DOA spectrum automatically, thus owns robustness to array imperfections. The main contributions of this paper are summarized as follows. 1) To the best of our knowledge, we first consider a framework that integrates the unfolded deep network into the DOA estimation by leveraging the spatial sparsity. To be specific, a well-understood iterative recovery algorithm, iterative shrinkage/thresholding algorithm, is unfolded to form a deep network with trainable variables. Due to the learning ability of the network, the sparsity of the incident signals and the array structure characteristics are acquired through the training procedure.

2) In order to alleviate the difficulty and complexity of sparse reconstruction, we exploit the sparsity embedded in the covariance vectors (columns of the covariance matrix) instead of original array outputs. This preprocessing step helps to reduce the size of the deep network and mitigate the requirements and expenses of training.

3) We provide performance analysis of the proposed deep learning method for DOA estimation in different cases. Specifically, we simulate normalized mean square error (NMSE) of the recovered spatial spectrum and root mean square error (RMSE) of the estimated signal directions to assess the recovery performance and DOA estimation accuracy. Also, extensive simulation results and comparison have further demonstrated the computation efficiency and robustness to array imperfections of the proposed scheme.

The rest of the paper consists of four parts. Section II reviews the overcomplete formulations of the array outputs and covariance vectors with and without array perturbations. Section III presents an unfolded deep learning framework–LISTA–and applies it to DOA estimation. Section IV carries out simulations to demonstrate the predominance of the proposed method . Section V concludes the whole paper.

II. Problem Formulation

A. Signal Model

Suppose that K independent far-field narrowband plane waves $\boldsymbol{s}(t)=[s_1(t),\, s_2(t),\ldots,s_K(t)]^{\mathrm{T}}$ with central frequency of ω impinge on an M-element array from directions $\boldsymbol{\theta}=[\theta_1,\theta_2,\ldots,\theta_K]$. The array output $\boldsymbol{x}(t)=[x_1(t),x_2(t),\ldots,x_M(t)]^{\mathrm{T}}$ at time t is:

$$\boldsymbol{x}(t)=\sum_{k=1}^{K}\boldsymbol{a}(\theta_k)s_k(t)+\boldsymbol{v}(t)=\boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{s}(t)+\boldsymbol{v}(t) \qquad (1)$$

where $\boldsymbol{A}(\boldsymbol{\theta})=[\boldsymbol{a}(\theta_1),\ldots,\boldsymbol{a}(\theta_K)]$ and $\boldsymbol{a}(\theta_k)=[1,e^{j\omega t_2(\theta_k)},\ldots,e^{j\omega t_M(\theta_k)}]^{\mathrm{T}}$ denotes the array steering matrix and vector respectively. $t_m(\theta_k)$ is the time-delay of the signal from direction θ_k from the first antenna to the mth antenna. $\boldsymbol{v}(t)$ represents the complex zero-mean Gaussian white noise vector with variance σ^2. Superscripts T and H denote transpose and Hermitian transpose, respectively.

In order to highlight the spatial sparsity of the incident signals, the array output $\boldsymbol{x}(t)$ should be reformulated in the following overcomplete form:

$$\boldsymbol{x}(t)=\sum_{l=1}^{L}\boldsymbol{a}(\vartheta_l)\overline{s}_l(t)+\boldsymbol{v}(t)=\boldsymbol{A}(\boldsymbol{\vartheta})\overline{\boldsymbol{s}}(t)+\boldsymbol{v}(t) \qquad (2)$$

where $\boldsymbol{\vartheta}=[\vartheta_1,\vartheta_2,\ldots,\vartheta_L]$ is a discrete direction set sampled from the potential space of the incident signals, with $\Delta\vartheta$ denoting the sampling interval. $\overline{\boldsymbol{s}}(t)$ is the zero-padded extension of $\boldsymbol{s}(t)$ from $\boldsymbol{\theta}$ to $\boldsymbol{\vartheta}$, which satisfies

$$\overline{s}_l(t)=\begin{cases} s_k(t), if\ |\vartheta_l-\theta_k|\le\frac{\Delta\vartheta}{2} \\ 0, \qquad otherwise \end{cases} \qquad (3)$$

According to Eq.(3), $\overline{\boldsymbol{s}}(t)$ only has nonzero elements at the true source locations, i.e., $\overline{\boldsymbol{s}}(t)$ is sparse.

Assume that N snapshots are collected, i.e., $t=t_1,\ldots,t_N$, then the observation matrix $\boldsymbol{X}$ can be written as:

$$\boldsymbol{X}=[\boldsymbol{x}(t_1),\ldots,\boldsymbol{x}(t_N)]=\overline{\boldsymbol{A}}\,\overline{\boldsymbol{S}}+\boldsymbol{V} \qquad (4)$$

where $\overline{\boldsymbol{S}}=[\overline{\boldsymbol{s}}(t_1),\ldots,\overline{\boldsymbol{s}}(t_N)]$ and $\boldsymbol{V}=[\boldsymbol{v}(t_1),\ldots,\boldsymbol{v}(t_N)]$ denotes the signal matrix and the noise matrix respectively. $\overline{\boldsymbol{A}}=\boldsymbol{A}(\boldsymbol{\vartheta})$ denotes the known overcomplete dictionary. The problem of DOA estimation can be solved by retrieving $\overline{\boldsymbol{S}}$ starting from the knowledge of $\boldsymbol{X}$ while the locations of peaks in the estimated $\hat{\overline{\boldsymbol{S}}}$ represent the signal directions.

B. Sparsity in Correlation Matrix

For static target, the spatial power spectrum $\boldsymbol{\eta}$ is also sparse with the same sparsity structure, i.e., $\boldsymbol{\eta}$ only has nonzero elements at the true source locations.

$$\boldsymbol{\eta}=[\eta_1,\eta_2,\ldots,\eta_L]^{\mathrm{T}},\eta_l=E[\overline{s}_l(t)\overline{s}_l^{\mathrm{H}}(t)] \qquad (5)$$

where $E(\bullet)$ stands for the expectation operator. The introduction of $\boldsymbol{\eta}$ simplifies the reconstruction procedure since the problem of retrieving $\overline{\boldsymbol{S}}$ is replaced by estimating $\boldsymbol{\eta}\in\mathbb{R}^{L\times 1}$.

The spatial correlation matrix $\boldsymbol{R}=E[\boldsymbol{x}(t)\boldsymbol{x}^{\mathrm{H}}(t)]$ is highly correlated to $\boldsymbol{\eta}$ and its mth column can be rewritten as:

$$\boldsymbol{z}_m=\boldsymbol{R}\boldsymbol{e}_m\triangleq\boldsymbol{W}_m\boldsymbol{\eta}+\sigma^2\boldsymbol{e}_m \qquad (6)$$

where $\boldsymbol{W}_m=[\boldsymbol{a}(\vartheta_1)\boldsymbol{a}^{\mathrm{H}}(\vartheta_1)\boldsymbol{e}_m,\ldots,\boldsymbol{a}(\vartheta_L)\boldsymbol{a}^{\mathrm{H}}(\vartheta_L)\boldsymbol{e}_m]$. $\boldsymbol{e}_m$ is an $M\times 1$ vector with the mth element being 1 and others being 0. Then, a new measurement vector $\boldsymbol{z} \in \mathbb{C}^{M^2 \times 1}$ can be defined by stacking the columns of $\boldsymbol{R}$ one-after-another:

$$\boldsymbol{z} = [\boldsymbol{z}_1; \ldots; \boldsymbol{z}_M] = \mathrm{vec}(\boldsymbol{R}) = \boldsymbol{W}\boldsymbol{\eta} + \sigma^2 \boldsymbol{e} \tag{7}$$

where $\boldsymbol{W} = [\boldsymbol{W}_1; \ldots; \boldsymbol{W}_M] \in \mathbb{C}^{M^2 \times L}, \boldsymbol{e} = [\boldsymbol{e}_1; \ldots; \boldsymbol{e}_M] \in \mathbb{R}^{M^2 \times 1}$. $[\bullet; \ldots; \bullet]$ operator stacks the arrays or vectors in sequence vertically. vec($\bullet$) vectorizes a matrix by stacking the columns of the matrix below each other.

In practical applications, the covariance matrix can only be estimated using N snapshots, so the estimated covariance vector $\hat{\boldsymbol{z}}$ is contaminated with estimation error vector $\Delta\boldsymbol{z}$ due to finite sampling:

$$\hat{\boldsymbol{z}} = \mathrm{vec}(\boldsymbol{R}) + \mathrm{vec}(\Delta\boldsymbol{R}) = \boldsymbol{W}\boldsymbol{\eta} + \Delta\boldsymbol{z} + \sigma^2 \boldsymbol{e} \tag{8}$$

where $\Delta\boldsymbol{R}$ is the estimation error matrix. Eq.(8) indicates that after removing $\sigma^2\boldsymbol{e}$, the estimated covariance vector $\hat{\boldsymbol{z}}$ is a noisy linear measurement of $\boldsymbol{\eta}$, with $\boldsymbol{W}$ representing the linear measurement operator and $\Delta\boldsymbol{z}$ denoting the additive noise item. Consequently, retrieving $\boldsymbol{\eta}$ from $\hat{\boldsymbol{z}}$ can be thought as a sparse linear inverse problem which could be solved by optimizing the following objective function:

$$L(\boldsymbol{\eta}) = \|\hat{\boldsymbol{z}} - \sigma^2\boldsymbol{e} - \boldsymbol{W}\boldsymbol{\eta}\|_2^2 + \gamma g(\boldsymbol{\eta}) \tag{9}$$

where $g(\boldsymbol{\eta})$ is a concave, non-decreasing penalty function, and γ is the regularization constant. $\| \bullet \|_2$ denotes the l_2 norm of a vector.

Eq.(8) differs from the typical sparse inverse problems [18], [19] in two folds: first, the commonly used assumption that the noise item is white Gaussian is invalid since different elements in $\Delta\boldsymbol{z}$ are correlated [14]. Second, $\hat{\boldsymbol{z}}$ has an extra perturbation item $\sigma^2\boldsymbol{e}$, which should be removed before recovery to maintain the sparsity profile of $\boldsymbol{\eta}$, so the noise variance estimation procedure should be implemented first.

C. Array Imperfections

The performance of model-driven sparse recovery method highly depends on the accuracy of the sensing matrix $\boldsymbol{W}$ and inaccurate modelling will significantly jeopardize the recovery results, hence decreasing the DOA estimation precision. However, various imperfections exist in practical sensor arrays, such as antenna pattern errors, gain and phase inconsistences, sensor position errors, mutual coupling, etc. These imperfections cause the array steering matrix deviates from $\boldsymbol{A}(\boldsymbol{\vartheta})$ to $\boldsymbol{A}^{\boldsymbol{p}}(\boldsymbol{\vartheta})$ where $\boldsymbol{p}$ denotes the perturbation parameters, and thus, sensing matrix $\boldsymbol{W}$ is changed to $\boldsymbol{W}^{\boldsymbol{p}}$. Consequently, the sparse linear model in Eq.(8) should be modified as follows,

$$\hat{\boldsymbol{z}}^{\boldsymbol{p}} = \boldsymbol{W}^{\boldsymbol{p}}\boldsymbol{\eta} + \Delta\boldsymbol{z}^{\boldsymbol{p}} + \sigma^2\boldsymbol{e} \tag{10}$$

Where $\hat{\boldsymbol{z}}^{\boldsymbol{p}}$,$\Delta\boldsymbol{z}^{\boldsymbol{p}}$ and $\boldsymbol{W}^{\boldsymbol{p}}$ denotes the estimated covariance vector, covariance estimation error vector and sensing matrix in the presence of imperfections, respectively. For model-driven methods such as Sparse Bayesian Learning (SBL) [14] and ISTA [23], one need a precise formulation for $\boldsymbol{W}^{\boldsymbol{p}}$ in order to achieve satisfactory recovery results and DOA estimation. Unfortunately, different kinds of array perturbations exert different influences on the array steering matrix $\boldsymbol{A}(\boldsymbol{\vartheta})$, and it is almost impossible to figure out a precise formulation for $\boldsymbol{A}^{\boldsymbol{p}}(\boldsymbol{\vartheta})$ in practical situations.

III. Proposed method

A. The Unfolded Iterative Shrinkage/Thresholding Algorithm

Recently, the advancement of deep learning has been exploited in designing algorithms for communications and signal processing. These methods provide a good combination of model-driven algorithms and the ability of networks to learn from training data, so it owns interpretability and robustness to array imperfections at the same time. We firstly give a brief introduction to the adopted algorithm to be unfolded, i.e. iterative shrinkage/thresholding algorithm (ISTA) (Algorithm 1). As it is originally designed for real-valued problems, we reformulate Eq.(8) into the following equation:

$$\breve{\hat{\boldsymbol{z}}} = \breve{\boldsymbol{W}}\boldsymbol{\eta} + \Delta\breve{\boldsymbol{z}} + \sigma^2\breve{\boldsymbol{e}} \tag{11}$$

Where $\breve{\hat{\boldsymbol{z}}} = \begin{bmatrix} \mathcal{R}(\hat{\boldsymbol{z}}) \\ \mathcal{I}(\hat{\boldsymbol{z}}) \end{bmatrix} \in \mathbb{R}^{2M^2 \times 1}$, $\breve{\boldsymbol{W}} = \begin{bmatrix} \mathcal{R}(\boldsymbol{W}) \\ \mathcal{I}(\boldsymbol{W}) \end{bmatrix} \in \mathbb{R}^{2M^2 \times L}$, $\Delta\breve{\boldsymbol{z}} = \begin{bmatrix} \mathcal{R}(\Delta\boldsymbol{z}) \\ \mathcal{I}(\Delta\boldsymbol{z}) \end{bmatrix} \in \mathbb{R}^{2M^2 \times 1}$, $\breve{\boldsymbol{e}} = \begin{bmatrix} \boldsymbol{e} \\ \boldsymbol{0} \end{bmatrix} \in \mathbb{R}^{2M^2 \times L}$. $\mathcal{R}(\bullet)$ and $\mathcal{I}(\bullet)$ represent the real and imaginary operators, respectively. $\boldsymbol{0}$ is an $M \times 1$ zero vector.

```
Algorithm 1 LISTA
Initialize $\hat{\sigma}^2, \{\varepsilon_t\}_{t=0}^{\mathrm{T}}, \{\boldsymbol{G}_t\}_{t=0}^{\mathrm{T}}, \{\boldsymbol{H}_t\}_{t=1}^{\mathrm{T}}$
$\boldsymbol{y} = \breve{\hat{\boldsymbol{z}}} - \hat{\sigma}^2\breve{\boldsymbol{e}}$,$\hat{\boldsymbol{\eta}}_0 = h(\boldsymbol{G}_0\boldsymbol{y}, \varepsilon_0)$
For $t = 1$ to $T$
$\boldsymbol{b}_t = \boldsymbol{G}_t\boldsymbol{y} + \boldsymbol{H}_t\hat{\boldsymbol{\eta}}_{t-1}$,$\hat{\boldsymbol{\eta}}_t = h(\boldsymbol{b}_t, \varepsilon_t)$
end
Return $\hat{\boldsymbol{\eta}}_T$
```

In Algorithm 1, we remove the noise component $\hat{\sigma}^2\breve{\boldsymbol{e}}$ from $\breve{\hat{\boldsymbol{z}}}$ before the reconstruction process. The noise variance is initialized according to $\hat{\sigma}^2 = \frac{1}{M-\hat{K}} \sum_{m=\hat{K}+1}^{M} \lambda_m$ where λ_m is the mth largest eigenvalue of $\hat{\boldsymbol{R}}$, $\hat{K}$ is the source number estimate obtained with MDL or AIC [24], [25]. $h(\bullet)$ represents an element-wise nonlinear transform which is defined as $h(\boldsymbol{b}_t, \varepsilon_t) = \mathrm{sign}(\boldsymbol{b}_t) * \max\{|\boldsymbol{b}_t| - \varepsilon_t, \boldsymbol{0}\}$ where * representing the element-wise Hadamard product. $\mathrm{sign}(\boldsymbol{b}_t)$ returns a vector containing the sign of each element in $\boldsymbol{b}_t$.

The original ISTA uses predefined and fixed $\boldsymbol{G}$, $\boldsymbol{H}$ and ε at all layers, while the LISTA use layer-dependent thresholds and matrices $\{\varepsilon_t\}_{t=0}^{\mathrm{T}}, \{\boldsymbol{G}_t\}_{t=0}^{\mathrm{T}}, \{\boldsymbol{H}_t\}_{t=1}^{\mathrm{T}}$ and ”optimize” these parameters from the training data $\{(\breve{\hat{\boldsymbol{z}}}^q, \boldsymbol{\eta}^q)\}_{q=1}^{Q}$ by minimizing the quadratic loss:

$$L_T(\boldsymbol{\Theta}) = \frac{1}{Q} \sum_{q=1}^{Q} \|\hat{\boldsymbol{\eta}}_T(\breve{\hat{\boldsymbol{z}}}^q, \boldsymbol{\Theta}) - \boldsymbol{\eta}^q\|_2^2 \tag{12}$$

where Q represents the size of the training dataset. $\boldsymbol{\Theta} = \{\{\varepsilon_t\}_{t=0}^{\mathrm{T}}, \{\boldsymbol{G}_t\}_{t=0}^{\mathrm{T}}, \{\boldsymbol{H}_t\}_{t=1}^{\mathrm{T}}\}$ denotes the set of learnable parameters, $\hat{\boldsymbol{\eta}}_T(\breve{\hat{\boldsymbol{z}}}^q, \boldsymbol{\Theta})$ is the output of the T-layer network with input $\hat{\boldsymbol{z}}^q$ and parameters $\boldsymbol{\Theta}$. Parameters $\boldsymbol{\Theta}$ can be updated based on the back-propagated gradients of the loss function with respect to the variables.

$$\mathbf{\Theta}^{new} = \mathbf{\Theta}^{old} + \mu_1 \frac{\partial L_T(\mathbf{\Theta})}{\partial \mathbf{\Theta}} \tag{13}$$

where μ_1 is the learning rate.

Making parameters $\mathbf{\Theta}$ trainable is one essential difference between the unfolded deep network and the original model-driven algorithm. This procedure enables the network to get competitive or even better performance with less layers. After training, the feed-forward process of the network consists only simple low-complexity calculations, making this structure more computationally efficient. Moreover, when array imperfections exist, $\mathbf{\Theta}$ will adapt to the imperfections through training, providing the network with robustness to array imperfections.

B. DOA estimation

Once we obtain $\hat{\boldsymbol{\eta}}_T$ through trained LISTA network, corresponding DOA can be estimated through peak searching. Since the predefined direction set $\boldsymbol{\vartheta}$ is formed via discrete spatial sampling, notable quantization errors may be introduced into the DOA estimates if the peak locations of $\hat{\boldsymbol{\eta}}_T$ are taken as the source directions directly. Therefore, we introduce a refined linear interpolation process to improve the DOA estimation precision based on the reconstruction result.

Take the kth peak cluster as an example. Suppose that it contains two significant magnitudes $\alpha_{k,1}$, $\alpha_{k,2}$ with the corresponding directions $\theta_{k,1}$ and $\theta_{k,2}$. Then, the estimate of the final DOA is calculated by:

$$\hat{\theta}_k = \frac{\alpha_{k,1}}{\alpha_{k,1} + \alpha_{k,2}} \theta_{k,1} + \frac{\alpha_{k,2}}{\alpha_{k,1} + \alpha_{k,2}} \theta_{k,2} \tag{14}$$

IV. NUMERICAL INVESTIGATION

We now investigate the performance of LISTA through a sequence of experiments on synthetic data. A typical reconstruction algorithm, SBL, is chosen to compete with LISTA.

A. Details of Experiments

A 10-element uniform linear array (ULA) with half-wavelength inter-element spacing is used. The interested spatial scope [-60°, 60°] is sampled with interval $\Delta\vartheta = 1°$ to form $L = 121$ grids, i.e., $\boldsymbol{\eta} \in \mathbb{R}^{121 \times 1}$. The training dataset is generated as follows: for each sample, its signal number K is set to 2 and the corresponding directions are randomly sampled in the scope of [-60°, 60°]. N=256 snapshots are collected. The signal matrix $\boldsymbol{S}$ is zero padded to $\overline{\boldsymbol{S}}$ according to Eq.(3), then the desired spectrum $\boldsymbol{\eta}$ is obtained by

$$\boldsymbol{\eta} = \frac{1}{N} diag(\overline{\boldsymbol{S}}\, \overline{\boldsymbol{S}}^{\mathrm{H}}) \tag{15}$$

where $diag(\bullet)$ gets the diagonal vector of a matrix. The corresponding measurement vector $\breve{\tilde{\boldsymbol{z}}}$ is generated according to Eq.(6)(7). We generate a total of 20000 samples $\{(\breve{\tilde{\boldsymbol{z}}}, \boldsymbol{\eta})\}$, 80% of which are used for training and the rest for validating. The parameters $\mathbf{\Theta}$ of LISTA are initialized as:

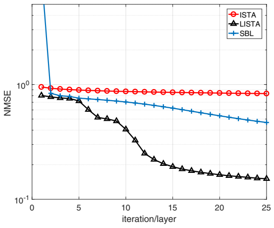


Fig. 1: Test NMSE vs layer (or vs iteration for ISTA and SBL)

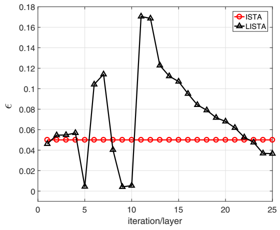


Fig. 2: RMSE vs angle separation

TABLE I: Averaged reconstruction time

Methods	LISTA	ISTA	SBL
Test time	9.113e-4s	9.123e-4s	0.15311s

$$\boldsymbol{G}_t = \frac{0.9}{\|\breve{\boldsymbol{W}}\|_2^2} \breve{\boldsymbol{W}}^{\mathrm{T}}, \boldsymbol{H}_t = \boldsymbol{I}_L - \frac{0.9}{\|\breve{\boldsymbol{W}}\|_2^2} \breve{\boldsymbol{W}}^{\mathrm{T}} \breve{\boldsymbol{W}} \tag{16}$$
$$\varepsilon_t = 0.05, t = 0, \ldots, T$$

To avoid bad local minimum, we employ the layer-by-layer training strategy as suggested in [19]. Our recent implementation is based on Tensorflow. Adam optimizer is used. The covariance vectors are used for training with mini-batch size of 500 and the order of the vectors is shuffled in each epoch. Early stopping strategy is adopted. The test platform is a PC with one Intel i7-8550U CPU.

B. Effect of Parameter Learning

We first study the effect of parameter learning by comparing the performance of LISTA and ISTA, which differ only in the use of parameter learning. We use the reconstruction normalized mean square error (NMSE) as the evaluation criteria.

$$\mathrm{NMSE} = \frac{1}{P} \sum_{i=1}^{P} \frac{\|\hat{\boldsymbol{\eta}}_T^i - \boldsymbol{\eta}^i\|_2^2}{\|\boldsymbol{\eta}^i\|_2^2} \tag{17}$$

where P is the size of the validation dataset. Figure 1 displays averaged test-NMSE versus layer t after 4000 realizations of $(\breve{\tilde{\boldsymbol{z}}}, \boldsymbol{\eta})$. The results show LISTA significantly outperforming both ISTA and SBL due to its learning procedure. Figure 2 shows the learned threshold ε_t of each layer compared with the initial value. The results show that LISTA learns layer-dependent thresholds $\{\varepsilon_t\}_{t=0}^{\mathrm{T}}$ instead of using a fixed value over all layers. This makes LISTA owns faster converging speed and superior adaptability than ISTA.

To assess the computational complexity of the proposed LISTA scheme, we record the time needs for very methods in TABLE I. These data are averaged for 3000 tests. Table I shows that the proposed LISTA scheme consumes two orders of magnitude less time than the current SBL algorithm. This superiority makes LISTA scheme especially suitable for applications where real time DOA estimation is needed.

We also give an intuitive view of the reconstructed spectra $\hat{\boldsymbol{\eta}}_T$ and the DOA estimation results of LISTA and compares the results with that of ISTA and SBL when no array imperfection is considered. Due to limitations of space, detailed descriptions are omitted.

C. Array imperfection adaptation

As has been discussed in Section I and III, learning based DOA estimation methods are expected to have built-in adaptations to typical array imperfections. We validate such property of the proposed method in this subsection.

Without loss of generality, three typical kinds of array imperfections – gain/phase inconsistence, sensor location error and inter-sensor mutual coupling – are considered in the simulations, and the array geometry is assumed to be linear to simplify notations. The proposed LISTA strategy can be generalized straightforwardly to other array geometries and imperfections if needed. To facilitate simulation, we use simplified models to formulate these perturbations. The simplification is perceived to be reasonable and fair for performance comparison since the proposed method makes no use of this prior information about the perturbations formulation.

When mutual coupling is present, the steering vector is $\boldsymbol{a}^{\boldsymbol{p}}(\boldsymbol{\vartheta}) = \boldsymbol{C}_{mc}\boldsymbol{a}(\boldsymbol{\vartheta})$, with $\boldsymbol{C}_{mc}$ representing the mutual coupling matrix. For ULA, $\boldsymbol{C}_{mc}$ can be written as

$$\boldsymbol{C}_{mc} = \text{Toeplitz}\left([1, b_1, \ldots, b_{M-1}]^{\mathrm{T}}\right) \tag{18}$$

where b_m is the coupling coefficient between two sensors separated by $m-1$ array elements. For gain/phase inconsistence, the steering vector is $\boldsymbol{a}^{\boldsymbol{p}}(\boldsymbol{\vartheta}) = \boldsymbol{C}_{g/p}\boldsymbol{a}(\boldsymbol{\vartheta})$ with the gain-phase matrix being

$$\boldsymbol{C}_{g/p} = \text{Diag}\left([\alpha_1 e^{j\phi_1}, \ldots, \alpha_M e^{j\phi_M}]^{\mathrm{T}}\right) \tag{19}$$

where $\alpha_1, \ldots, \alpha_M$ and $\phi_1, \ldots, \phi_M$ are the gains and initial phases of the sensors. Toeplitz($\bullet$) forms a Toeplitz matrix by taking the given vector as the first column and Diag($\bullet$) forms diagonal matrices with the given vector on the diagonal. In the presence of sensor location error, the array responding vector is $\boldsymbol{a}^{\boldsymbol{p}}(\boldsymbol{\vartheta}) = \boldsymbol{a}_{le}(\boldsymbol{\vartheta}) * \boldsymbol{a}(\boldsymbol{\vartheta})$, with $\boldsymbol{a}_{le}(\boldsymbol{\vartheta})$ denoting the location error vector and * representing the element-wise Hadamard product. For ULA with inter-sensor spacing of d, $\boldsymbol{a}(\boldsymbol{\vartheta})$ and $\boldsymbol{a}_{le}(\boldsymbol{\vartheta})$ can be written more explicitly as

$$\begin{aligned} \boldsymbol{a}(\boldsymbol{\vartheta}) &= [1, \ldots, e^{j\omega(M-1)dsin(\vartheta)/c}]^{\mathrm{T}} \\ \boldsymbol{a}_{le}(\boldsymbol{\vartheta}) &= [e^{j\omega d_1' sin(\vartheta)/c}, \ldots, e^{j\omega d_M' sin(\vartheta)/c}]^{\mathrm{T}} \end{aligned} \tag{20}$$

with $d_1', \ldots d_M'$ being the sensor location errors. c represents the propagation speed of the signal. In the paper, we take the first array sensor as the reference, thus $\phi_1 = 0$,$\alpha_1 = 1$,and $d_1' = 0$.

In the simulation, the mutual coupling coefficients, the initial sensor gains/phases and the sensors location errors are generated according to

$$b_m = \mu_{mc}\left(0.6\rho\text{exp}(j\rho\frac{\pi}{2})\right)^m, m = 1, \ldots, M-1 \tag{21}$$

$$\begin{aligned} \alpha_m &= \mu_{g/p}(1 + 0.3\rho\zeta_m) \\ \phi_m &= \mu_{g/p}\rho\zeta_m\frac{\pi}{4}, m = 2, \ldots, M \end{aligned} \tag{22}$$

$$d_m' = \mu_{le}0.5\rho\zeta_m d, m = 2, \ldots, M \tag{23}$$

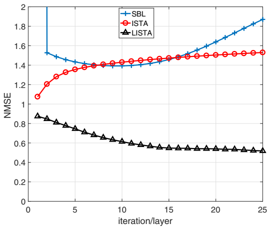


Fig. 3: Test NMSE versus layer with array imperfections

where the parameter $\rho \in [0, 1]$ is introduced to control the strength of the imperfections, When $\rho = 0$, no imperfection is contained in the array responding functions. $\mu_\bullet$ is used to indicate whether a certain kind of imperfection exists. $\zeta_\bullet$ is set to positive or negative 1 with equal probability in order to add randomness to array perturbations. With above notifications, the perturbed array responding function is rewritten as follows

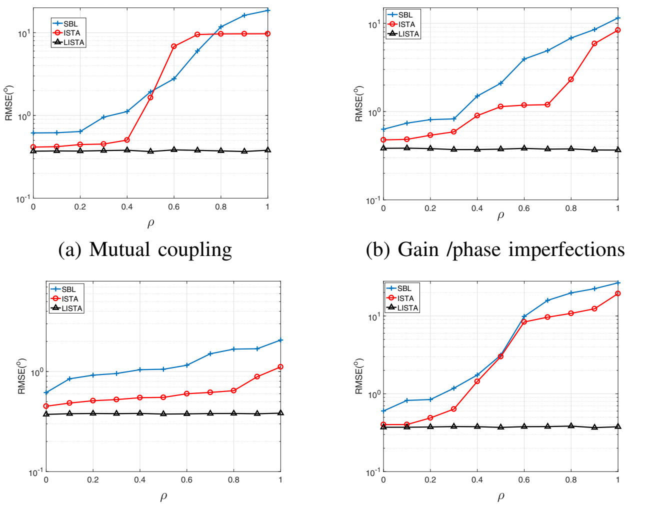
$$\boldsymbol{a}^{\boldsymbol{p}}(\boldsymbol{\vartheta}) = \boldsymbol{C}_{g/p}\boldsymbol{C}_{mc}\left(\boldsymbol{a}_{le}(\boldsymbol{\vartheta}) * \boldsymbol{a}(\boldsymbol{\vartheta})\right) \tag{24}$$

Firstly, we repeat the experiment in section IV-B to prove the robustness of the proposed method to array perturbations. The most severe situation where $\rho = 1$, $\mu_{mc} = \mu_{g/p} = \mu_{le} = 1$ is considered. The experiment conditions are the same with Figure 1 except that the array outputs are contaminated with array perturbations. SBL and LISTA are not calibrated since we have no prior information about the array imperfections. Figure 3 displays average test-NMSE versus layer t after 4000 simulations. The results show that ISTA and SBL do not converge since they use an incorrect sensing matrix $\boldsymbol{W}$. LISTA remains satisfactory performance because it can learn the sensing matrix adaptively from the training data.

Then, we evaluate the effect of different array imperfections furtherly. We choose the ROOT Mean Square Error (RMSE) as the quantitative index to evaluate the DOA estimation precision.

$$\text{RMSE} = \sqrt{\frac{1}{HK}\sum_{i=1}^{\mathrm{H}} \|\hat{\boldsymbol{\theta}}^i - \boldsymbol{\theta}\|^2} \tag{25}$$

where $\hat{\boldsymbol{\theta}}^i$ is the estimation results in the ith test, $\boldsymbol{\theta}$ is the true signal direction set, H is times of the Monte-Carlo simulations and K is the number of signals. Two signals with SNR=0dB are assumed to impinge onto the array from directions of -15.5°and 13.6°. Four cases are considered by setting $\mu_\bullet$ to different values in Eq.(21)(22)(23). First, mutual coupling effect is considered by setting $\mu_{mc} = 1, \mu_{g/p} = \mu_{le} = 0$. The direction estimation RMSE versus ρ averaged by 3000 simulations are shown in Fig.4(a). Then we set $\mu_{g/p} = 1, \mu_{mc} = \mu_{le} = 0$ to consider gain /phase imperfections and $\mu_{le} = 1, \mu_{mc} = \mu_{g/p} = 0$ to retain the sensor position error, with the corresponding results shown in Fig.4(b) and (c). At last, the combined effect of all the three kinds of array imperfections are evaluated and shown in Fig.4(d) by setting $\mu_{mc} = \mu_{g/p} = \mu_{le} = 1$. The results show that as the array imperfections become more and more significant,



(a) Mutual coupling

(b) Gain /phase imperfections

(c) Sensor position error

(d) Combined array inconsistence

Fig. 4: RMSE in the presence of different array imperfections

DOA estimation errors of ISTA and SBL increase significantly. Therefore, additional calibration procedure must be applied to improve the estimation precision. On the contrary, the proposed learning based method performs robustly to different and combined array imperfections, verifying its superiority in non-ideal scenarios.

V. Conclusion

In this paper, an efficient deep unfolded network based spatial spectra recovery algorithm is proposed and applied to DOA estimation. This approach is obtained by unfolding the ISTA algorithm into a deep network and learning the network parameters that best fit the training dataset. Compared with conventional hand designed algorithms, the proposed DOA estimation method achieves robustness to various array imperfection due to its learning capability. Moreover, the feed-forward process of the network consists only simple calculations which greatly speeds up the DOA estimation procedure. A set of numerical experiments verified the superiority of the proposed method clearly. Simulation results show it adapts well to noisy training data, off-grid incident signals, even various kinds of array imperfections.

References

[1] S. Kikuchi, H. Tsuji, and A. Sano, "Autocalibration algorithm for robust capon beamforming," *IEEE Antennas and Wireless Propagation Letters*, vol. 5, pp. 251–255, 2006.

[2] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[3] F. Bellili, C. Elguet, S. B. Amor, S. Affes, and A. Stphenne, "Code-aided doa estimation from turbo-coded qam transmissions: Analytical crlbs and maximum likelihood estimator," *IEEE Transactions on Wireless Communications*, vol. 16, no. 5, pp. 2850–2865, 2017.

[4] J. Dai, X. Xu, and D. Zhao, "Direction-of-arrival estimation via real-valued sparse representation," *IEEE Antennas and Wireless Propagation Letters*, vol. 12, pp. 376–379, 2013.

[5] L.-l. Wu, Z.-m. Liu, and W.-l. Jiang, "A direction finding method for spatial optical beam-forming network based on sparse bayesian learning," *Signal, Image and Video Processing*, vol. 11, no. 2, pp. 203–209, Feb 2017.

[6] P. Xiao, B. Liao, and N. Deligiannis, "Deepfpc: A deep unfolded network for sparse signal recovery from 1-bit measurements with application to doa estimation." *Signal Processing*, vol. 176, p. N.PAG, 2020.

[7] L. Wan, Y. Sun, L. Sun, Z. Ning, and J. J. P. C. Rodrigues, "Deep learning based autonomous vehicle super resolution doa estimation for safety driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4301–4315, 2021.

[8] L. Wu and Z. Huang, "Coherent svr learning for wideband direction-of-arrival estimation," *IEEE Signal Processing Letters*, vol. 26, no. 4, pp. 642–646, 2019.

[9] Z. Liu, C. Zhang, and P. S. Yu, "Direction-of-arrival estimation based on deep neural networks with robustness to array imperfections," *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 12, pp. 7315–7327, 2018.

[10] H. Huang, J. Yang, Y. Song, H. Huang, and G. Gui, "Deep learning for super-resolution channel estimation and doa estimation based massive mimo system," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 1–1, 2018.

[11] B. Chen, H. Xiang, M. Yang, and C. Li, "Altitude measurement based on characteristics reversal by deep neural network for vhf radar," *IET Radar, Sonar and Navigation*, vol. 13, no. 1, pp. 98–103, 2019.

[12] Z. Huang, L. Wu, and Z. Liu, "Towards wide-frequency-range direction finding with support vector regression," *IEEE Communications Letters*, pp. 1–1, 2019.

[13] L.-L. Wu, Z.-M. Liu, and Z.-T. Huang, "Deep convolution network for direction of arrival estimation with sparse prior," *IEEE Signal Processing Letters*, vol. 26, no. 11, pp. 1688–1692, 2019.

[14] Z. Liu and Y. Zhou, "A unified framework and sparse bayesian perspective for direction-of-arrival estimation in the presence of array imperfections," *IEEE Transactions on Signal Processing*, vol. 61, no. 15, pp. 3786–3798, 2013.

[15] T. Lo, H. Leung, and J. Litva, "Radial basis function neural network for direction-of-arrivals estimation," *IEEE Signal Processing Letters*, vol. 1, no. 2, pp. 45–47, 1994.

[16] A. Randazzo, M. A. Abou-Khousa, M. Pastorino, and R. Zoughi, "Direction of arrival estimation based on support vector regression: Experimental validation and comparison with music," *IEEE Antennas and Wireless Propagation Letters*, vol. 6, pp. 379–382, 2007.

[17] S. Vigneshwaran, N. Sundararajan, and P. Saratchandran, "Direction of arrival (doa) estimation under array sensor failures using a minimal resource allocation neural network," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 2, pp. 334–343, 2007.

[18] K. Gregor and Y. Lecun, "Learning fast approximations of sparse coding," 08 2010, p. 399406.

[19] M. Borgerding, P. Schniter, and S. Rangan, "Amp-inspired deep networks for sparse linear inverse problems," *IEEE Transactions on Signal Processing*, vol. PP, pp. 1–1, 05 2017.

[20] D. Ito, S. Takabe, and T. Wadayama, "Trainable ista for sparse signal recovery," *IEEE Transactions on Signal Processing*, vol. 67, no. 12, pp. 3113–3125, 2019.

[21] H. K. Aggarwal, M. P. Mani, and M. Jacob, "Modl: Model-based deep learning architecture for inverse problems," *IEEE Transactions on Medical Imaging*, vol. 38, no. 2, pp. 394–405, 2019.

[22] B. Hu, X. Wu, X. Zhang, Q. Yang, and W. Deng, "Off-grid doa estimation based on compressed sensing with gain/phase uncertainties," *Electronics Letters*, vol. 54, no. 21, pp. 1241–1243, 2018.

[23] A. Chambolle, R. A. De Vore, Nam-Yong Lee, and B. J. Lucier, "Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 319–335, 1998.

[24] A.-K. Seghouane, "Asymptotic bootstrap corrections of aic for linear regression models," *Signal Processing*, vol. 90, pp. 217–224, 01 2010.

[25] P. Stoica and Y. Selen, "Model-order selection: a review of information criterion rules," *IEEE Signal Processing Magazine*, vol. 21, no. 4, pp. 36–47, July 2004.