DIRECT LOCALIZATION IN UNDERWATER ACOUSTICS VIA CONVOLUTIONAL NEURAL NETWORKS: A DATA-DRIVEN APPROACH

SUPPLEMENTARY MATERIALS

Amir Weiss, Toros Arikan and Gregory W. Wornell

Massachusetts Institute of Technology

A. PROOF OF PROPOSITION 1

For convenience, we first define

$$\widetilde{C}_{\text{SBL}}(\boldsymbol{p},\bar{\boldsymbol{s}},\boldsymbol{B}) \triangleq \sum_{\ell=1}^{L} \|\bar{\boldsymbol{x}}_\ell - \boldsymbol{H}_\ell(\boldsymbol{p},\mathcal{E})\bar{\boldsymbol{s}}\|_2^2 . \tag{S1}$$

Using the identity $\mathrm{Diag}\left(\boldsymbol{D}_\ell \boldsymbol{b}_\ell\right)\bar{\boldsymbol{s}} = \overline{\boldsymbol{S}}\boldsymbol{D}_\ell\boldsymbol{b}_\ell$, it is easily seen that, for every $\ell \in \{1,\dots,L\}$, $\widetilde{C}_{\text{SBL}}(\boldsymbol{p},\bar{\boldsymbol{s}},\boldsymbol{B})$ of (S1) is minimized with respect to $\boldsymbol{b}_\ell$ by

$$\widehat{\boldsymbol{b}}_\ell = \left(\left(\overline{\boldsymbol{S}}\boldsymbol{D}_\ell\right)^{\mathrm{H}}\overline{\boldsymbol{S}}\boldsymbol{D}_\ell\right)^{-1}\left(\overline{\boldsymbol{S}}\boldsymbol{D}_\ell\right)^{\mathrm{H}}\bar{\boldsymbol{x}}_\ell, \tag{S2}$$

assuming[1] $\mathrm{rank}\left(\boldsymbol{D}_\ell\right) = 3$ for all $\ell \in \{1,\dots,L\}$ hereafter. Substituting $\widehat{\boldsymbol{B}} \triangleq [\widehat{\boldsymbol{b}}_1 \cdots \widehat{\boldsymbol{b}}_L]$ into $\widetilde{C}_{\text{SBL}}(\boldsymbol{p},\bar{\boldsymbol{s}},\boldsymbol{B})$ yields

$$\begin{aligned}\breve{C}(\boldsymbol{p},\bar{\boldsymbol{s}})_{\text{SBL}} &\triangleq \widetilde{C}_{\text{SBL}}(\boldsymbol{p},\bar{\boldsymbol{s}},\widehat{\boldsymbol{B}}) = \sum_{\ell=1}^{L}\left\|\bar{\boldsymbol{x}}_\ell - \overline{\boldsymbol{S}}\boldsymbol{D}_\ell\widehat{\boldsymbol{b}}_\ell\right\|_2^2\\ &= \sum_{\ell=1}^{L}\big[\bar{\boldsymbol{x}}_\ell^{\mathrm{H}}\bar{\boldsymbol{x}}_\ell - \bar{\boldsymbol{x}}_\ell^{\mathrm{H}}\overline{\boldsymbol{S}}\boldsymbol{D}_\ell\widehat{\boldsymbol{b}}_\ell - \widehat{\boldsymbol{b}}_\ell^{\mathrm{H}}\left(\overline{\boldsymbol{S}}\boldsymbol{D}_\ell\right)^{\mathrm{H}}\bar{\boldsymbol{x}}_\ell\\ &\qquad + \widehat{\boldsymbol{b}}_\ell^{\mathrm{H}}\left(\overline{\boldsymbol{S}}\boldsymbol{D}_\ell\right)^{\mathrm{H}}\overline{\boldsymbol{S}}\boldsymbol{D}_\ell\widehat{\boldsymbol{b}}_\ell\big].\end{aligned} \tag{S3}$$

From (S2), we observe that

$$\widehat{\boldsymbol{b}}_\ell^{\mathrm{H}}\left(\overline{\boldsymbol{S}}\boldsymbol{D}_\ell\right)^{\mathrm{H}}\overline{\boldsymbol{S}}\boldsymbol{D}_\ell\widehat{\boldsymbol{b}}_\ell = \widehat{\boldsymbol{b}}_\ell^{\mathrm{H}}\left(\overline{\boldsymbol{S}}\boldsymbol{D}_\ell\right)^{\mathrm{H}}\bar{\boldsymbol{x}}_\ell,$$

with which (S3) simplifies to

$$\breve{C}_{\text{SBL}}(\boldsymbol{p},\bar{\boldsymbol{s}}) = \underbrace{\sum_{\ell=1}^{L}\bar{\boldsymbol{x}}_\ell^{\mathrm{H}}\bar{\boldsymbol{x}}_\ell}_{\substack{\text{constant with respect to}\\ \boldsymbol{p} \text{ and } \bar{\boldsymbol{s}}}} - \sum_{\ell=1}^{L}\bar{\boldsymbol{x}}_\ell^{\mathrm{H}}\overline{\boldsymbol{S}}\boldsymbol{D}_\ell\widehat{\boldsymbol{b}}_\ell. \tag{S4}$$

Code available at https://www.weissamir.com/project/DLOC.

[1] We ignore the extreme, unrealistic cases in which $\boldsymbol{D}_\ell$ are not full rank, which rarely occur only for very specific settings of the receivers' and source's positions. Nonetheless, the initial optimization is performed via a grid search, hence we can discard points giving rise to these rare settings.

Therefore, using (S4), we may now simplify (5) by

$$\min_{\substack{\bar{\boldsymbol{s}}\in\mathcal{S}_N\\ \boldsymbol{B}\in\mathbb{C}^{3\times L}}} \widetilde{C}_{\text{SBL}}(\boldsymbol{p},\bar{\boldsymbol{s}},\boldsymbol{B}) = \max_{\bar{\boldsymbol{s}}\in\mathcal{S}_N}\sum_{\ell=1}^{L}\bar{\boldsymbol{x}}_\ell^{\mathrm{H}}\overline{\boldsymbol{S}}\boldsymbol{D}_\ell\widehat{\boldsymbol{b}}_\ell. \tag{S5}$$

At this point, notice that using

$$\bar{\boldsymbol{x}}_\ell^{\mathrm{H}}\overline{\boldsymbol{S}} = \bar{\boldsymbol{s}}^{\mathrm{T}}\overline{\boldsymbol{X}}_\ell^{*} \Longrightarrow \left(\overline{\boldsymbol{S}}\boldsymbol{D}_\ell\right)^{\mathrm{H}}\bar{\boldsymbol{x}}_\ell = \left(\overline{\boldsymbol{X}}_\ell^{\mathrm{H}}\boldsymbol{D}_\ell\right)^{\mathrm{H}}\bar{\boldsymbol{s}}^{*},$$

we may write

$$\widehat{\boldsymbol{b}}_\ell = \left(\boldsymbol{D}_\ell^{\mathrm{H}}\overline{\boldsymbol{S}}^{\mathrm{H}}\overline{\boldsymbol{S}}\boldsymbol{D}_\ell\right)^{-1}\boldsymbol{D}_\ell^{\mathrm{H}}\overline{\boldsymbol{X}}_\ell\bar{\boldsymbol{s}}^{*}. \tag{S6}$$

Substituting $\widehat{\boldsymbol{b}}_\ell$ from (S6) into (S5), using (S6), $\overline{\boldsymbol{S}}^{\mathrm{H}}\overline{\boldsymbol{S}} = \mathrm{Diag}(|\bar{\boldsymbol{s}}|^2) \triangleq \boldsymbol{P}_{\bar{\boldsymbol{s}}}$ and simplifying further yields

$$\begin{aligned}&\max_{\bar{\boldsymbol{s}}\in\mathcal{S}_N}\sum_{\ell=1}^{L}\bar{\boldsymbol{x}}_\ell^{\mathrm{H}}\overline{\boldsymbol{S}}\boldsymbol{D}_\ell\widehat{\boldsymbol{b}}_\ell =\\ &\max_{\bar{\boldsymbol{s}}\in\mathcal{S}_N}\sum_{\ell=1}^{L}\bar{\boldsymbol{s}}^{\mathrm{T}}\overline{\boldsymbol{X}}_\ell^{*}\boldsymbol{D}_\ell\left(\boldsymbol{D}_\ell{}^{\mathrm{H}}\boldsymbol{P}_{\bar{\boldsymbol{s}}}\boldsymbol{D}_\ell\right)^{-1}\boldsymbol{D}_\ell^{\mathrm{H}}\overline{\boldsymbol{X}}_\ell\bar{\boldsymbol{s}}^{*} = \qquad \text{(S7)}\\ &\max_{\bar{\boldsymbol{s}}\in\mathcal{S}_N}\bar{\boldsymbol{s}}^{\mathrm{H}}\left(\sum_{\ell=1}^{L}\overline{\boldsymbol{X}}_\ell\boldsymbol{D}_\ell^{*}\left(\boldsymbol{D}_\ell{}^{\mathrm{T}}\boldsymbol{P}_{\bar{\boldsymbol{s}}}\boldsymbol{D}_\ell^{*}\right)^{-1}\left(\overline{\boldsymbol{X}}_\ell\boldsymbol{D}_\ell^{*}\right)^{\mathrm{H}}\right)\bar{\boldsymbol{s}},\end{aligned}$$

$$\text{(S8)}$$

where from (S7) to (S8) we have used that $\boldsymbol{P}_{\bar{\boldsymbol{s}}} \in \mathbb{R}_{+}^{N\times N}$, and that (S7) is real-valued (and nonnegative).

By assumption, $\boldsymbol{P}_{\bar{\boldsymbol{s}}} = \rho^2 \cdot \boldsymbol{I}_N$. Thus, in this case (S8) simplifies further to

$$\begin{aligned}&\max_{\bar{\boldsymbol{s}}\in\mathcal{S}_N}\bar{\boldsymbol{s}}^{\mathrm{H}}\left(\sum_{\ell=1}^{L}\overline{\boldsymbol{X}}_\ell\boldsymbol{D}_\ell^{*}\left(\boldsymbol{D}_\ell^{\mathrm{T}}\boldsymbol{D}_\ell^{*}\right)^{-1}\left(\overline{\boldsymbol{X}}_\ell\boldsymbol{D}_\ell^{*}\right)^{\mathrm{H}}\right)\bar{\boldsymbol{s}} =\\ &\max_{\bar{\boldsymbol{s}}\in\mathcal{S}_N}\bar{\boldsymbol{s}}^{\mathrm{H}}\boldsymbol{Q}(\boldsymbol{p},\mathcal{E})\bar{\boldsymbol{s}} = \lambda_{\max}\left(\boldsymbol{Q}(\boldsymbol{p},\mathcal{E})\right),\end{aligned}$$

where $\boldsymbol{Q}(\boldsymbol{p},\mathcal{E})$ is defined in (6). Therefore, we conclude that when $|\bar{s}[k]| = \rho$, the SBL position estimate is given by

$$\widehat{\boldsymbol{p}}_{\text{SBL}} = \underset{\boldsymbol{p}\in\mathbb{R}^{3\times 1}}{\mathrm{argmax}}\;\lambda_{\max}\left(\boldsymbol{Q}(\boldsymbol{p},\mathcal{E})\right).$$

B. ALTERNATIVE FORM OF THE SBL OBJECTIVE

Define the Cholesky decompositions

$$\boldsymbol{D}_\ell^{\mathrm{T}} \boldsymbol{D}_\ell^* \triangleq \boldsymbol{\Gamma}_\ell^{\mathrm{H}} \boldsymbol{\Gamma}_\ell \in \mathbb{C}^{R\times R},\ \forall \ell \in \{1,\dots,L\},$$

where $\boldsymbol{\Gamma}_\ell \in \mathbb{C}^{R\times R}$, and further define

$$\boldsymbol{U}(\boldsymbol{p},\mathcal{E}) \triangleq \left[\overline{\boldsymbol{X}}_1 \boldsymbol{D}_1^* \boldsymbol{\Gamma}_1^{-1} \cdots\ \overline{\boldsymbol{X}}_L \boldsymbol{D}_L^* \boldsymbol{\Gamma}_L^{-1}\right] \in \mathbb{C}^{N\times RL}.$$

Let

$$\widetilde{\boldsymbol{Q}}(\boldsymbol{p},\mathcal{E}) \triangleq \boldsymbol{U}(\boldsymbol{p},\mathcal{E})^{\mathrm{H}} \boldsymbol{U}(\boldsymbol{p},\mathcal{E}) \in \mathbb{C}^{RL\times RL}. \tag{S9}$$

Based on [1, Prop. 3], we have

$$\lambda_{\max}\left(\boldsymbol{Q}(\boldsymbol{p},\mathcal{E})\right) = \lambda_{\max}\left(\widetilde{\boldsymbol{Q}}(\boldsymbol{p},\mathcal{E})\right), \tag{S10}$$

which, when written explicitly with

$$\boldsymbol{G}_\ell \triangleq \boldsymbol{D}_\ell^* \boldsymbol{\Gamma}_\ell^{-1} \in \mathbb{C}^{N\times R}, \tag{S11}$$

gives the required alternative form.

C. THE ACCOMPANYING CODE PACKAGE

The supplementary materials of this paper contain a code package with the implementation of the proposed data-driven CNN-based DLOC method, as well as the implementation of the competing model-based localization methods considered in Section 5, namely, GCC-PHAT, SBL and MFP.

The code package contains two folders: ‘Python’ and ‘Matlab’. Our CNN-based DLOC is implemented in Python, and all of its associated files are in the ‘Python’ folder. All the other methods are implemented in Matlab, and their associated files are in the ‘Matlab’ folder.

D. CONTENTS OF THE PYTHON FOLDER

This folder contains a single file, ‘requirements.txt’, which contains the names of the required Python packages, and four sub-folders:

1) ‘DOA’: This folder contains two files:
   - `DOA_model_MCE.py`: The model for direction-of-arrival (DOA) estimation; and
   - `train_DOA_model_script_MLSP2022.py`: Script for training the model for DOA estimation.
2) ‘Elevation’: This folder contains two files:
   - `Elevation_model_MCE.py`: The model for elevation (/inclination) estimation; and
   - `train_Elevation_model_script_MLSP2022.py`: Script for training the model for elevation estimation.
3) ‘Range’: This folder contains two files:
   - `range_model_MSE.py`: The model for range estimation; and
   - `train_range_model_script_MLSP2022.py`: Script for training the model for range estimation.
4) ‘Direct_Localization’: This folder contains seven files:
   - `direct_localization_model_weights_spherical.py`: The model for DLOC, which includes all the sub-models mentioned above, and the custom loss functions;
   - `train_direct_localization_model_script_MLSP2022.py`: Script for training the DLOC model. Note that for this training procedure, weights of the trained sub-models for range, DOA and elevation estimation are required for initialization (as explained in the paper);
   - `script_for_testing_the_trained_model_MLSP2022.py`: Script for testing the trained DLOC model;
   - `direct_localization_model_static.index` and `direct_localization_model_static.data-00000-of-00001`: Weights for the “static” DLOC model; and
   - `direct_localization_model_dynamic.index` and `direct_localization_model_dynamic.data-00000-of-00001`: Weights for the “dynamic” DLOC model.

E. CONTENTS OF THE MATLAB FOLDER

The ‘Matlab’ folder contains four files:

- `compute_GCCPHAT_cost_function_efficient.m`: Computes the GCC-PHAT cost function;
- `compute_oracleMFP3_cost_function.m`: Computes the “oracle” (known attenuation coefficients) MFP cost function for the three-ray model;
- `compute_SBL_cost_function_efficient.m`: Computes the SBL cost function;
- `compute_SBL_and_oracleMFP3_objective_function.m`: Computes both the SBL and MFP cost functions;
- `MSE_vs_SNR_for_MLSP2022_static.m`: Script for generating the RMSE vs. the SNR for the setting considered in the paper for SBL, MFP and GCC-PHAT for the “static” scenario;
- `MSE_vs_SNR_for_MLSP2022_dynamic.m`: Script for generating the RMSE vs. the SNR for the setting considered in the paper for SBL, MFP and GCC-PHAT for the “dynamic” scenario.

F. REFERENCES

[1] A. Weiss, T. Arikan, H. Vishnu, G. B. Deane, A. C. Singer, and G. W. Wornell, “A semi-blind method for localization of underwater acoustic sources,” *IEEE Trans. Signal Process.*, vol. 70, pp. 3090–3106, 2022.