On the Semblance Based TDOA Algorithm for Sound Source Localization: a parametric study

Guilherme Seidyo Imai Aldeia, Henrique Ferreira, Kenji Nose Filho

***Abstract*— Recently proposed, the *Semblance Based TDOA Algorithm for Sound Source Localization* uses a cross-correlation function to determine the direction of a sound source. This article aims to perform a parametric analysis of the algorithm, applying a hypothesis test to determine the importance of each parameter. The results show that, regarding the three original parameters of the algorithm, one presented small sensitivity and can be discarded, and the remaining can be defined based on the desired resolution level, or fixed at low values (at cost of a slower runtime).**

***Keywords*— semblance, parametric analysis, parameter optimization, TDOA.**

I. Introduction

Many algorithms have parameters that can be previously adjusted in a given configuration which is believed to maximize their performance in a specific task. This is advantageous because it allows the user to perform fine-tuning of the algorithm, but also results in an additional effort to be used: the task of adjusting these parameters, which, in many cases, may require additional information, or may not be trivial.

The simplest way to select parameters is to use *ad-hoc* values, which can be based on other values seen in the literature or on a heuristic process. Another way to make this selection is to use some parameter adjustment method, such as *gridsearch* or *manual search* [1].

The parameter adjustment step aims to improve the efficiency of the algorithm when solving the problem and can be divided into two complementary methodologies: optimization and parametric analysis. The optimization consists of finding a set of values for the parameters that make the algorithm more efficient in relation to a cost function, being, therefore, a criterion for maximizing performance. The parametric analysis consists of determining whether the alteration of a given parameter influences the result of the algorithm and how much the result is sensitive to the variation of that parameter (sensitivity analysis), being, in turn, a criterion of economy of the variables used in the model. Since numerical optimization is, in general, a computationally expensive process [2], using a parametric analysis allows to search, in a constrained space of the parameters, relationships of the parameters and their relation with the overall performance, then use the results to get insights of the behavior of the algorithm in function to the configurations, with the expectation that the analysis leads to an elimination of parameters with low sensitivity or which not influence significantly the algorithm performance.

Guilherme Seidyo Imai Aldeia is from Center of Mathematics, Computing and Cognition, Federal University of ABC, e-mail: guilherme.aldeia@ufabc.edu.br; Henrique Ferreira and Kenji Nose Filho are from Center of Engineering, Modeling and Applied Social Sciences, Federal University of ABC, e-mail: {hfsantos, kenji.nose}.ufabc.edu.br.

Recently, an algorithm for the sound source localization (SSL) task was proposed, by using a cross-correlation function common in seismic signal processing [3], [4], called *Semblance Based TDOA Algorithm for Sound Source Localization* (SB-SSL) [5]. The SSL problem is strongly dependent of the environmental conditions during audio capture (such as noise, echo, and interference), as well as the dynamic nature of the sound sources (if they are moving in relation to the capture system and if they produce constant, intermittent or sudden sounds). To deal with this variability of scenarios, SB-SSL was proposed with three parameters that can be adjusted by the user.

Our objectives are to analyze the parameters of the SB-SSL algorithm and check if *(i)* there is a subset of parameters that can have a fixed value without impacting performance, *(ii)* evaluate how the method's accuracy varies when each parameter is modified, and *(iii)* to draw attention to the importance and benefits that sensitivity analysis can provide when using a computational method.

Our contributions are the elaboration of a methodology to analyze the influence of different parameters (that can be extended to other algorithms), the application of the proposed methodology in a study of case of the SB-SSL algorithm, and a systematic way of manipulating audios to build a data set.

The remaining of this paper is organized as follows. In Section II, the target algorithm of the study is briefly presented, with explanations to its parameters. In Section III, the methodology for evaluating the parameters is presented, along with the data used for the work. In Section IV, the numerical results obtained are presented, and in Section V, this results are discussed. Finally, in Section VI, the conclusion about the parameters is made.

II. Semblance Based TDOA algorithm

The SB-SSL algorithm is based on an array of k microphones ($k \geq 2$) as a signal capture and input tool, spaced at an appropriate distance, which is related on the sampling rate, so that a sound signal — which is assumed to have a flat wavefront — can be captured by the microphones at different time stamps. The spatial arrangement of the microphones plays an important role in disambiguation of the estimated location (i.e. if the array of microphones are organized in a linear form, it will not be able to determine the exact elevation, as both positive and negative elevations would have the same correlation).

The determination of the sound source direction is done by creating a uniformly spaced grid of possible direction values for the sound source then, for each direction, a delay related to the position of the channel in the microphone array is applied to each channel of the signal. Taking as a possible direction $\mathbf{k}_d(\Theta_d, \Phi_d) \in \mathbb{R}^3$, which can be parameterized by azimuth $\Theta_d \in [-\pi, \pi]$ and elevation $\Phi_d \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, the corresponding delay for the microphone located at $\mathbf{m}_k \in \mathbb{R}^3$, using as a reference point $\mathbf{0} = [\mathbf{0}, \mathbf{0}, \mathbf{0}]^\mathbf{T}$ [6], is calculated by:

$$\tau_{d,k} = -\frac{\mathbf{k}_d \cdot \mathbf{m}_k}{v}, \tag{1}$$

where v is the speed of sound, and $\cdot$ denotes the internal product operation.

Then, the Semblance Coherence function Z_d is calculated, which determines a metric of energy relationship between the channels. For each pair (Θ_d, Φ_d) in a uniformly spaced grid, with spacing given by the Δ parameter, Z_d is calculated by:

$$Z_d = \frac{\sum_n \left|\sum_k \hat{s}_k(n)\right|^2}{N_r \sum_n \sum_k \left|\hat{s}_k(n)\right|^2}, \tag{2}$$

where k denotes the microphones, n denotes the time samples, N_r is the total number of sensors, $\hat{s}_k(n) = s_k(n - \tau_{d,k})$ is the signal at time n of the k-th microphone after the time correction $\tau_{d,k}$.

By combining the correlation measures for each tested direction, a matrix of coherence values is created, called the Semblance Panel. The row and column of the element with the highest value in this matrix indicates the values of (Θ_d, Φ_d) for which there is maximum consistency between the signals received in each microphone.

Although a single Semblance Panel can be obtained from an arbitrarily sized audio, it was observed that splitting the audio into windows provide a higher accuracy [5]. In this case, the set of panels obtained are recombined using the *max pooling* method, where each value of the final panel at position (θ_i, ϕ_j) is the maximum value at the same position among all panels obtained for each window.

In the original paper [5], the authors present three parameters that can be adjusted, presented below with their findings:

- **Delta (Δ)** - scan step to generate the uniformly spaced grid. This value determines the maximum angular resolution that the search grid can have. The best value obtained was $10°$, and in general, smaller values did not present a numerically significant difference in the result;
- **FrameSize (w)** - the size of the windows that the algorithm will use for each attempt to estimate the sound source — during single data acquisition, several windows are created, the semblance panel is generated for each one and, finally, the various windows are recombined by *max pooling*. The best value was $0.064s$ (the lowest value allowed in their analysis), and the authors concluded that the lower the value for this parameter, the better the result;
- **Overlap (δ)** - the size of the overlap between each window, causing consecutive windows to share part of the signals that were used to compute the correlation in the adjacent and underlying windows. A degree of 20% overlap had better results.

Despite that the authors had performed a *gridsearch* process to determine the best parameter values, the work had a limited database, not covering several possible scenarios.

III. Methodology

This work uses two databases: a validation base and a test base. The validation base was used to obtain results from each possible configuration allowed during the process of *gridsearch*, and is composed of 150 audios synthetically generated; and a test base, which contains 300 real audios, recorded with a drone, used to obtain results on non-artificial data. The domain problem of our data sets remained the same as the original paper [5] — finding a speech source using signals from an 8-microphone array attached to a drone.

As a measure of performance, the Equation 3 — *Great Circle Distance* (GCD) [7] — was used, which computes the smallest angle between two points on the surface of a sphere, where θ_1 and θ_2 represents the real and predicted azimuth, ϕ_1 and ϕ_2 the real and predicted elevation, and $\Delta\theta$ is equal to the absolute difference of θ_1 and θ_2.

Lastly, a hypothesis test was also performed between the results obtained, in order to determine which significant differences should be considered during the discussion of the results. The next items describe in detail each mentioned aspect of the methodology.

A. Validation and Test Data

We used data from the *DREGON data set* [8], in which sets of recordings made with an arrangement of 8 microphones coupled to a drone are available. They provide a set of 300 audios of speech containing ego-noise (from drone rotors) and a set of 3 audios of speech without ego-noise, both sets for static sources. Further, they provided a set of audio with only ego-noise for different rotors velocity. The set of 300 audios was used only in the test partition, to obtain results that correspond to the performance in a real scenario. As we observed that the original data presents a low variety of directions from the sound source, being able to skew the model and favor some specific configurations, we increase the number of directions in the 3-audios set performing an audio synthesis stage. It was necessary to create a set of audios for validation — that was achieved by creating synthetic audios based on the combination of a set of noise-free audios and a set of pure ego-noise audios made with the same equipment and applying a reverse TDOA model to make new source positions for each created audio. Thus, the validation partition have 150 audios, quantity that allows obtaining significantly large samples for the application of hypothesis tests, while considerably decreasing the algorithm validation time for each possible configuration. Figure 1 shows the directions of the two sets. Note that while the validation set have 150 audios, the real data partition (test) contains 300 audios but with only 12 different directions.

For the synthetic audio generation, a combination was made between 3 noise-free audios, with an active speech source and a known true direction; and pure noise recordings of single drone rotors at different speeds. The speech audios already had

$$\Delta\sigma = \arctan\left(\frac{\sqrt{(\cos\phi_2\sin(\Delta\theta))^2 + (\cos\phi_1\sin\phi_2 - \sin\phi_1\cos\phi_2\cos(\Delta\theta))^2}}{\sin\phi_1\sin\phi_2 + \cos\phi_1\cos\phi_2\cos(\Delta\theta)}\right). \quad (3)$$

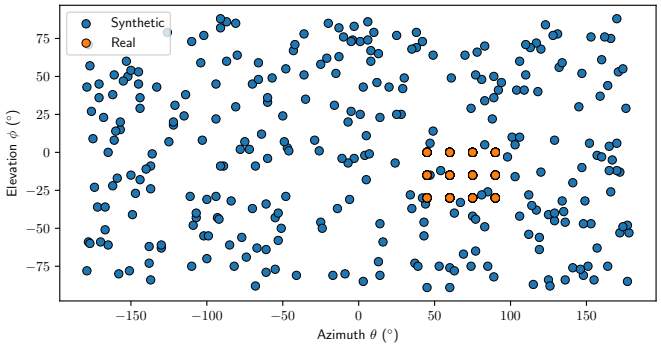

Fig. 1. True directions of synthetic and real data sets.

a direction, being necessary to align the signals before drawing a new random direction. For this, the channels go through a process of *upsample*, increasing the original sampling rate of 44.100 kHz by 4 times to provide smoothed signals when compensating the delays. After that, a random direction is drawn, the delay is computed for that corresponding direction, and then applied to the signals. Finally, the signal goes through a *downsample*, returning to its original sample rate, and is combined with 4 noises, each corresponding to the noise generated by one isolated rotor (the combination of the 4 noises represents the 4 rotors running at different speeds), where each rotor has its speed drawn at random, and the noise is cut at a random point for each noise channel. To combine the speech signal and the noise, both are normalized and a randomly selected *signal-to-noise ratio* (SNR) between [-25, 12]. All random draws are made with equal probabilities, in an attempt to create a uniformly distributed database. As a result, there is no direct control over the directions and relationships between signal and noise on the generated data.

B. Gridsearch

The *gridsearch* is a simple algorithm optimization technique, easy to implement and capable of giving better results than manual optimization, and is reliable in small search spaces, although in larger search spaces the *random search* has better performance [1]. The random search is able to find more precise values, not fixed to the grid defined by the possible values that the parameters are allowed to assume. Considering that this work is trying to analyse a small number of parameters, a *gridsearch* is satisfactory, as it has a reasonable execution time.

First, we defined possible values that each of parameter can assume, summarized in Table I. These values were determined so that the range with the lowest values was covered with higher resolution than the range with the highest values, where it is believed that the best settings are concentrated, taking as reference the original paper [5].

The *gridsearch* process will evaluate, for each possible configuration, the mean error over the validation audios. Finally, the configuration that presents the best configuration will be performed for the test partition as a sanity check.

TABLE I
POSSIBLE CONFIGURATIONS FOR EACH PARAMETER.

Parameter	Range
Δ	$\{5, 7.5, 15, 30\}$ degrees (°)
w	$\{0.064, 0.128, 0.256, 0.512, 1.024\}$ seconds (s)
δ	$\{10, 20, 30, 40, 50\}$ %

C. Parametric Analysis

The parametric analysis will be done by comparing the results over the validation data set using the *Wilcoxon Signed-Ranked* hypothesis test between related samples. It is a non-parametric hypothesis test, not requiring assumptions about the behavior of the sample distribution [9], in contrast to parametric tests, such as the *Student's t-test* [10], which assumes that the samples come from a normal distribution. The *Wilcoxon test* estimates the acceptance of the null hypothesis (that the difference in the distribution means is given at random) or the alternative hypothesis (that there is, in fact, a statistically relevant difference between the samples), where the returned *p-value* is used as an indicator of acceptance or rejection of the null hypothesis. In the literature, a *p-value* ≤ 0.05 is used as a strong indicator over the alternative hypothesis.

The hypothesis test will be done to provide statistical evidence when there is or is not a statistical difference between the observed results for different parameter configurations of SB-SSL. Although alternative methods rather than the *Wilcoxon test* have been proposed, this test is still widely used in the literature and has implementations available in several programming languages.

IV. RESULTS

Considering the possible values for each parameter, in total there are 125 configurations. The *gridsearch* created and evaluated each configuration with the validation data, with their average error reported in Figure 2, with the best configuration underlined. Once the best configuration was determined, all the others were tested by the *Wilcoxon test* and those that had a *p-value* greater than 0.05 were marked with an asterisk.

Notice that, for the best configuration, the parameters assumes the values $\Delta = 7.5°$, $w = 0.064s$, and $\delta = 0.5\%$, with an average error of $12.84°$. Running the algorithm with this configuration for the test data, the average error obtained is 15.86, being reasonable to argue that the synthetic data presents robustness and captures features similar to the real audios.

To check the relationships between the best and worst configurations, Figure 3 presents a graph known as *radar plot* (or *spider plot*), used to visualize multidimensional data, where

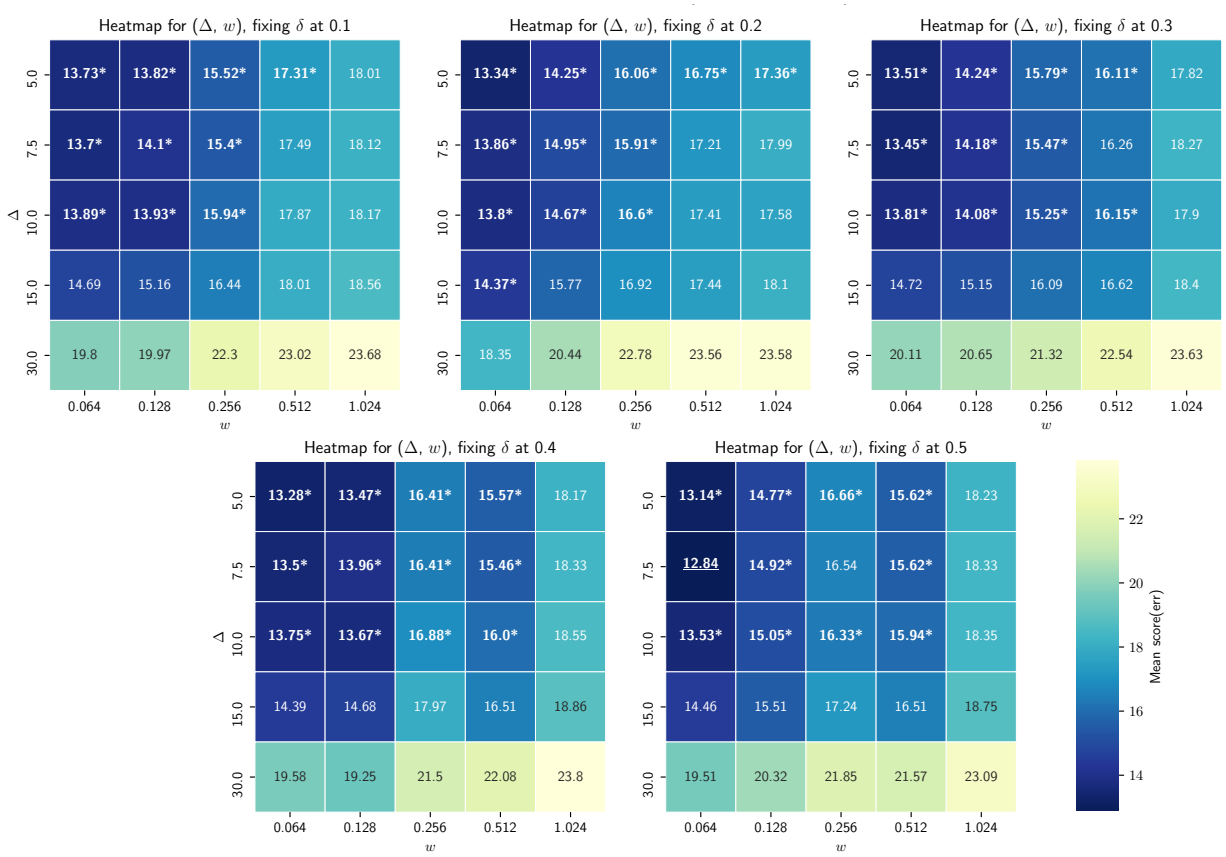

Fig. 2. Heatmaps sharing the same colorscale. The best configuration is hightlighted underlined, and every other configuration that had a *p-value*> 0.05 are marked with an asterisk.

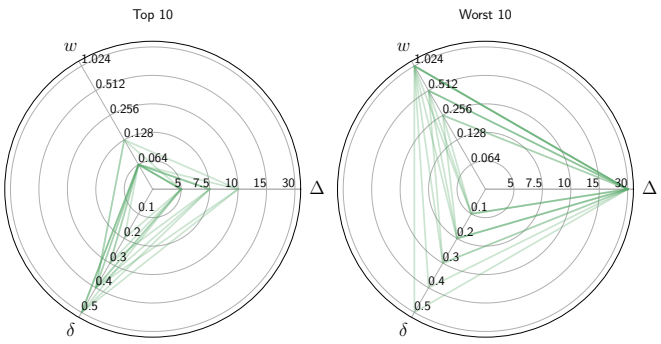

Fig. 3. Radar plot grouping the 10 best and worst configurations.

TABLE II

MEAN PERCENTAGE OF THE PERFORMANCE CHANGE AS A SINGLE PARAMETER IS VARIED, INCREMENTED ITERATIVELY THROUGH EACH ALLOWED VALUE, WHERE THE ASTERISK INDICATES A *p-value* > 0.05.

Δ (smallest: 5.0)	7.5	10.0	15.0	30.0
Relative to previous value	-0.85927	-0.72547*	-4.0949	-30.87169
Relative to smallest value	-0.85927	-1.59097	-5.75102	-38.39815
w (smallest: 0.064)	0.128	0.256	0.512	1.024
Relative to previous value	-4.78447	-11.41239	-2.07177*	-8.31954
Relative to smallest value	-4.78447	-16.74288	-19.16152	-29.07522
δ (smallest: 0.1)	0.2	0.3	0.4	0.5
Relative to previous value	-0.10279*	1.75658	-0.12018*	-0.62796*
Relative to smallest value	-0.10279*	1.65559*	1.5374*	0.91909*

each axis represents a parameter with it's possible values, and a configuration is represented by a closed polygon that intercepting each axis at the values that the configuration assumes.

Yet, another way to measure performance is to pair all settings where only one parameter changes to the next allowed value — analysis *one-at-a-time* — and calculate the mean percentage of the increase or decrease in performance. As the settings are paired, the statistical test can be applied here again. The results are reported in Table II.

V. DISCUSSION

Regarding the *heatmaps* in Figure 2, as there are three parameters, each *heatmap* corresponds to a slice of the cube that could be assembled for each tuple $(\Delta_i, w_j, \delta_k)$ corresponding to a configuration — by visualizing the *heatmaps* stacked, the *z-axis* represents the δ. Through *heatmaps* we can see that small values of Δ and w always dominate larger values, implying that the performance increases as those values decrease. Furthermore, considering the δ axis, cases with a high *p-value* are shared (in other words, through each *heatmap*, the configurations with high *p-values* are in

the same positions), but considering the rows/columns that would represent the Δ or w axes, the same does not occur, although it is possible to observe that almost all configurations with $\Delta = \{5.0, 7.5, 10.0\}$ presents a high *p-value*, implying that this parameter is relevant to determine the algorithm performance, but from a point on, it seems unjustifiable to decrease even further. For Δ, the dominance of small values can be explained, as it increases the spatial resolution and allows to estimate the direction with a smaller error. Also for w, considering that several windows are created, since the speech signal is not active at all times, and may be masked by noise in large windows, the dominance of low values is justified. On the other hand, it seems unreasonable to argue that δ renders any benefits, as it works by "recycling" signals that have already been processed, a functionality that apparently does not justify being used.

The *spider plots* in Figure 3 show that the best settings always share small values for Δ and w, but the parameter δ assumes all except the smallest value in the range. From this plot, given that the best solutions shares almost the same configurations for Δ and w, there is a strong indicator that those two parameters are critical to the performance. Similarly, the worst settings always have high values for Δ and w, and δ assumes all values allowed. In addition, it is possible to observe in *heatmaps* that almost all configurations with performance close to the best are in the region of low values for Δ and w.

Regarding the *p-value*, every configuration on the top 10 presented an asterisk on the *heatmaps*, indicating that this difference is given at random, instead of having a solid statistical evidence that they render different results. Although, when comparing the best and the worst configurations on *heatmap*, it is evident that none of them have an asterisk. The main difference on the *spider plots* of best and worst configurations leads to the conclusion that small values for Δ and w are a requisite to a better performance.

Analyzing the Table II with the percentage of performance variation, it is possible to see that there is a consecutive decrease in performance as Δ or w are increased, with the row relative to the lowest value representing the accumulation of performance gain. Only δ does not seem to define a pattern of performance increase or loss, having small variations and, generally, high *p-values*. It is visible that the increase of Δ and w parameters renders a drastically performance drop.

VI. Conclusion

In contrast to the original article [5], our results were obtained here with a larger data set, although almost all findings are preserved. While the contributions of the original article were the SB-SSL algorithm and an analysis of SNR level where the algorithm can perform well, our work investigated the parameters through the usage of statistical tests.

We can conclude that the parameter δ shows strong signs of being unnecessary, allowing to simplify the usage of the algorithm, in addition to simplifying the fine-tuning process, due to the smaller number of parameters to adjust.

In addition, as initially noted by the authors, small values of Δ and w show better results, although the lower the values, the higher the computational cost required, as it increases the grid resolution (controlled by Δ) or the number of times the grid is created (controlled by w — a small window causes a audio to be divided into a larger number of windows, being necessary to apply the algorithm in each one, and increasing the number of panels to be used in *max pooling*).

It seems to exist a threshold where minimizing Δ and w renders any benefits. Since they are strongly related to computational performance, an execution of the same experiments would help to choose the appropriate values for different application domains.

Thus, there is a clear compromise between resolution and execution time. However, it is possible to notice from the *heatmaps* that the settings, for the possible determined values, have errors in the range of $20 - 25°$, which can be considered satisfactory for some applications.

Based on the *gridsearch*, we were able to perform a parametric study at a relatively low computational cost and with a constrained search space, finding similar conclusions from the original paper, as well as extending it to a larger data set, resulting the proposal of removing δ parameter from SB-SSL algorithm.

References

[1] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(10):281–305, 2012.

[2] Mark French. *Fundamentals of Optimization: Methods, Minimum Principles and Applications for making Things Better*, pages 38–39. Springer, 2018.

[3] N. S. Neidell and M. T. Taner. Semblance and other coherency measures for multichannel data. *Geophysics*, 36(3):482–497, 1971.

[4] T. Barros, R. Lopes, and M. Tygel. Implementation aspects of eigendecomposition-based high-resolution velocity spectra. *Geophysical Prospecting*, 63(1):99–115, 2015.

[5] Guilherme Aldeia, Alex Crispim, Guilherme Barreto, Kaleb Alves, Henrique Ferreira, and Kenji Nose-Filho. A semblance based tdoa algorithm for sound source localization. In *XXXVII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*, 10 2019.

[6] J. Nikunen and T. Virtanen. Time-difference of arrival model for spherical microphone arrays and application to direction of arrival estimation. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 1255–1259, Aug 2017.

[7] T. Vincenty. Direct and inverse solutions of geodesic on the ellipsoid with application of nested equations. *Survey Review*, 23(176):88–93, 1975.

[8] Martin Strauss, Pol Mordel, Victor Miguet, and Antoine Deleforge. DREGON: Dataset and Methods for UAV-Embedded Sound Source Localization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2018)*, pages 5735–5742, Madrid, Spain, October 2018. IEEE.

[9] Frank Wilcoxon. *Individual Comparisons by Ranking Methods*, pages 196–202. Springer New York, New York, NY, 1992.

[10] Student. The Probable Error of a Mean. *Biometrika*, 6(1):1–25, 03 1908.