# A Simulation Tool for the Analysis of TDOA Methods

André B. de F. Diniz[1], Danilo de S. Pena[2] and Vicente A. de Sousa Jr.[1]

[1]Federal University of Rio Grande do Norte (UFRN), Lagoa Nova, Natal-RN, 59078-970 Brazil
[2]Sidia, Av. Darcy Vargas, 654, Parque Dez de Novembro, Manaus-AM, 69055-035, Brazil

**Abstract**

Localization applications are widely used as an essential service in many systems, such as 5G device positioning, location of things (LoT), and speech source localization. Unlike classical GPS, the time difference of arrival (TDOA) approach estimates the angle of arrival using the signals received from a sensor array, allowing to determine the position of an emitter with more accuracy. This paper presents the Simulation PlAtform for PosiTIon LoCatiOn (SimPatico), a simulation tool for evaluation and analysis of TDOA methods, providing a precise way to compare their performance considering specific scenarios. TDOA methods implemented in SimPatico are shown in terms of mathematical formulation and operational properties. We also explore the main features of the simulator, including source signal representations, addition of distortion on them and simulation parameters. Simulation outcomes are presented, exemplifying SimPatico as a benchmark platform and a versatile tool for performance analysis of TDOA methods.

**Keywords**

TDOA, Simulation, Algorithms, Localization

## 1. Introduction

The process of extracting information from a sensor array and its processing enables the localization of sources in a physical environment. These sensors are capable of monitoring sources, providing indoor location information for a variety of applications. Nowadays, where internet connectivity has increased massively, the devices might enjoy positioning services in their applications. Consequently, with the emerging technologies and the internet of things (IoT), the position information plays an essential role in allowing novel solutions in many different contexts.

One way to determine the indoor position is by the time difference of arrival (TDOA), a technique able to estimate the angle of arrival based on the delay between the signals received in the sensors. The TDOA has many possible uses in signal processing and telecommunication fields, for example, device positioning in 5G [1], positioning for vehicles [2], speech source localization systems [3], powerline fault localization [4], and target location systems [5]. However, comparison and evaluation of individual methods may be impractical for a specific application

*IPIN 2021 WiP Proceedings, November 29 – December 2, 2021, Lloret de Mar, Spain*
andrebfd@ufrn.edu.br (A. B. d. F. Diniz); danilo.pena@sidia.com (D. d. S. Pena); vicente.sousa@ufrn.edu.br (V. A. d. Sousa Jr.)
0000-0002-0694-4972 (A. B. d. F. Diniz); 0000-0003-2767-5764 (D. d. S. Pena); 0000-0003-2859-6136 (V. A. d. Sousa Jr.)

considering a variety of algorithms and diversity of setups to be examined. Moreover, the performance of the methods depends on the aspects of the signal and the scenario conditions, such as noise assumptions, robustness, and computational cost, making the solutions incomparable due to the hard reproducibility of such evaluations.

Simulating the efficiency of TDOA methods, setting parameters such as different source signals, the number of sensors and noise models, is essential to determine the more appropriate TDOA system and its configuration for an application. Thus, the development of a simulation tool that can simulate the effects of those algorithms for various schemes is crucial, giving the means to compare them and providing resources for distinct situations. It may offer means for designers to assess quantitatively and statistically the performance of the techniques, allowing them to reach a good balance of the accuracy-complexity trade-off of such methods.

## 2. Related Works

Simulation tools in the field of localization techniques are available, such as PerfLoc [6] and IndoorLoc [7]. However, those studies examine only indoor strategies, where they obtain the position information from GPS, via Wi-Fi, and the received signal strength indicator (RSSI). Although there is a relationship between source localization and TDOA, they are not equivalent. Therefore, those studies focus on data collecting, mapping and tracking. The simulator SMILe [8] has been proposed based on TDOA and time of flight (ToF), which provides an open-source means of evaluating a few methods with the capability of setting the clock synchronization. Nevertheless, the simulator does not evaluate distortion conditions and channel models and uses only the mean squared error (MSE) for performance evaluation.

Studies have been carried out on prototyping tools regarding asynchronous TDOA [9]. Although it is possible to perform comparisons between real and simulated data, this simulator is focused on hardware and does not allow a more sophisticated statistical analysis, such as probability of resolution and root-mean-squared error (RMSE). Another simulator was built based on chirp spread spectrum for wireless location purposes [10]. However, it employs few methods for TDOA calculations based on adaptive filtering and it is limited to signaling formatting as source signal.

Based on these needs, we developed the Simulation PlAtform for PosiTIon LoCatiOn evaluation (SimPatico). It features implementations of TDOA methods with weighting functions in the frequency domain. We also examine a variety of noise components, accounting for gaussian and non-gaussian models. These are essential in order to address a wide variety of physical phenomena regarding the methods in applications based on TDOA techniques. SimPatico also enables the user to adjust setup configurations and different benchmark processes by plotting their performances with regards to SNR, error levels, and other figures of merit.Thus, our contributions can be summarized as follows:

- Provide an open-source simulation tool, named SimPatico, to help in the conception and evaluation of TDOA methods;
- Provide a unified simulation platform to benchmark different TDOA methods;
- Provide a unified simulation platform to test TDOA algorithms in different impairment models, including gaussian and non-gaussian modeled noises, and reverberation effects;

- Provide the implementation of several TDOA and DOA methods with a user-friendly configuration platform;
- Provide plenty of performance comparison examples, including aspects such as (i) gaussian and impulsive noises; (ii) analysis of resolution and number of sensors ; (iii) reverberation analysis.

This paper is organized as follows. In Section 3, we describe the signal model and the TDOA methods implemented on the simulator, presenting their benefits and limitations. The channel modeling is presented in Section 4, with a brief discussion of their characteristics. Our simulator is shown in Section 5. In Section 6, the main results are presented and discussed, comparing the performance of the implemented TDOA algorithms using the simulator. In Section 7, we present our final remarks.

# 3. TDOA Principles

The TDOA problem consists of estimating the delay $\tau$ between signals arriving on a set of sensors. In a situation of an array with two sensors, for example, one can realize this procedure by calculating the cross-correlation between the two signals and determine $\tau$ by observing the moment where the resulting function reaches its maximum value. However, this approach may not be accurate due to the presence of environmental phenomena, such as noise and reverberation.

Therefore, estimating the TDOA of incoming signals using different approaches is essential to compare and decide the most efficient method for each scenario.

In the following, the utilized signal model for one source is given by:

$$r_i(n) = a_i x(n - \tau_i) + u_i(n), \tag{1}$$

where $a_i x(n - \tau_i)$ corresponds to scaled source signal received by the $i$th sensor, with a relative delay represented by $\tau_i$, and $u_i(n)$ denotes the noise components uncorrelated with $x(n)$ and each other.

In order to analyze the TDOA methods, their parameters and models must be taken into account, as well as the weighting functions. SimPatico contains a variety of methods for such calculations, which includes filters for generalized cross-correlation (GCC) procedures and the steered-response power phase transform (SRP-PHAT), as presented in the following sections.

## 3.1. Generalized Cross-Correlation (GCC)

One can analyze the cross-correlation between the incoming signals from a source, with data acquired by different sensors. The peak of the resulting function corresponds to the TDOA; however, its position may be affected by noise and reverberation [11]. In an attempt to minimize these limitations, it is possible to introduce a weighting function for the correlation process, which can be seen in the frequency domain as [12]:

$$R^{cc}_{x_1 x_2}(n) = \mathcal{F}^{-1}[X_1(\omega) X_2^*(\omega) \phi(\omega)], \tag{2}$$

where $\mathcal{F}^{-1}[\cdot]$ denotes the inverse Fourier transform, $X_1(\omega)$ corresponds to the first signal in the frequency domain and $X_2^*(\omega)$ to the complex conjugate of the second signal in the same domain, and their product with the Fourier inverse yields the correlation process; $\phi(\omega)$ is a pre-filter that implements weighting functions, which can assume different expressions. The GCC process is illustrated in Figure 1.

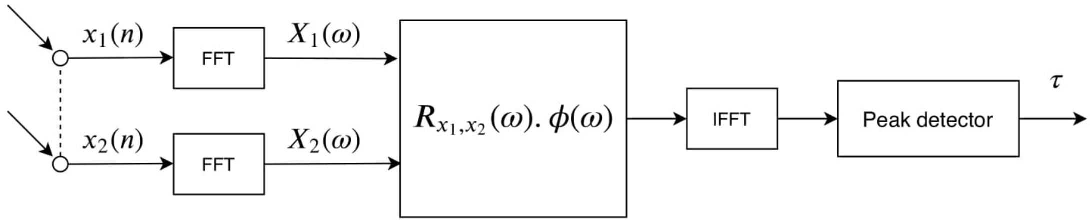



**Figure 1:** Flowchart of GCC.

Table 1 presents a list of weighting functions, including cases where noise characteristics may be known or not [12].

**Table 1**
TDOA weighting functions.

| Weighting function | $\phi(\omega)$ | Noise knowledge |
|---|---|---|
| Roth [13] | $\frac{1}{X_1(\omega)X_1^*(\omega)}$ | no |
| SCOT [14] | $\frac{1}{\sqrt{X_1(\omega)X_1^*(\omega)X_2(\omega)X_2^*(\omega)}}$ | no |
| PHAT [15] | $\frac{1}{\lvert X_1(\omega)X_2^*(\omega)\rvert}$ | no |
| Eckart [16] | $\frac{X_1(\omega)X_1^*(\omega)}{[s(\omega)s^*(\omega)]^2}$ | yes |
| MLE [17] | $\frac{\lvert\gamma(\omega)\rvert^2}{\lvert X_1(\omega)X_2^*(\omega)\rvert[1-\lvert\gamma(\omega)\rvert^2]}$ | no |
| MMLE [18] | $\frac{\lvert\gamma(\omega)\rvert}{\left[\lvert X_1(\omega)X_2^*(\omega)\rvert^{0.75}+\min(\gamma(\omega))\right]\sqrt{\lvert 1-\gamma(\omega)^2\rvert}}$ | no |

The signal $s(\omega)$ denotes the noise components in the frequency domain, and the Wiener-Hopf optimum filter $\gamma(\omega)$ is given by

$$\gamma(\omega) = \frac{X_1(\omega)X_2^*(\omega)}{\sqrt{X_1(\omega)X_1^*(\omega)X_2(\omega)X_2^*(\omega)}}. \tag{3}$$

Each weighting function accounts for different properties on TDOA estimation, producing different methods. The Roth filter suppresses frequency regions where the result of the correlation process may possess significant error [13]. Whereas, the smoothed coherent transform (SCOT) can be seen as a pre-whitening filter with a cross-correlation operation in the sequence [14], leading to a superior performance in the case of low noise level. On the other hand, the phase transform (PHAT) makes the cross-correlation operation solely dependent on the channel responses [19] by discarding the amplitude and keeping only the phase information. In particular, Eckart maximizes the deflection criterion, which is the ratio of change in mean correlation output due to the signal of interest compared to the component of noise solely. The maximum likelihood estimator (MLE) reaches the minimum possible variance for an unbiased estimator, also known as the Cramér-Rao Lower Bound. However, to achieve its best performance, the signals should not be affected by reverberation, the delay has to be constant, and the signals should be described as stationary processes [19]. Finally, a modified maximum likelihood estimator (MMLE) [18] is also present in SimPatico, which aims to achieve optimal performance in noisy and reverberant scenarios coherently. This can be reached because its formulation addresses some signal properties, such as the variance of the estimated parameter (the angle of arrival $\theta$) and the cross-spectrum phase.

### 3.2. Steered-Response Power Phase Transform

One of the most popular TDOA methods is based on maximizing the steered response power (SRP) in a beamforming process. The SRP-PHAT method can be seen as an extension of the GCC-PHAT, which corresponds to summing a complex-steered, weighted, and filtered version of the data signal received at each sensor [20]. The utilized filter in this context is the absolute value of the signal, corresponding to the PHAT weighting function, described in Table 1. Thus, the SRP-PHAT response function in the frequency domain is

$$R(\omega) = \sum_{n=1}^{M} \frac{X_n(\omega) X_{n+1}^*(\omega)}{|X_n(\omega) X_{n+1}^*(\omega)|} e^{j\omega\tau}, \tag{4}$$

where $M$ denotes the number of sensors. Given the mathematical representations of GCC-PHAT in Table 1 and SRP-PHAT in Eq. 4, it can be seen that in the presence of only two sensors ($M = 2$), these methods have the same performance.

## 4. Channel modeling

In real-world situations, one must account for imperfections in the channel where the analyzed signal travels through a certain medium. There are many phenomena involving physical systems, so the TDOA methods must be modeled appropriately to reach accurate estimates. Therefore, the present simulator contains a variety of functions in order to account for these effects, which are channel reverberation and background noise. For the latter, the available additive noise in the signal are gaussian and impulsive noises (modeled by gaussian mixture or $\alpha$-stable models).

### 4.1. Gaussian Model

It is widely known that additive white gaussian noise (AWGN) is commonly found in telecommunication applications, following a gaussian distribution. Its probability density function (PDF) is modeled by the following random variable $x$ as:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \tag{5}$$

where $\mu$ represents the mean value and $\sigma^2$ the variance. Although very popular, the gaussian model is not able to model heavy tails distributions since its tail has an exponential shape.

### 4.2. Gaussian Mixture Model

The gaussian mixture model (GMM) is a linear combination of a series of gaussian PDFs and may be used for modeling impulsive noises [21]. Its model can be represented by

$$p(y) = \sum_{i=1}^{M} c_i \frac{1}{\sigma_i \sqrt{2\pi}} e^{-(x_i-\mu_i)^2/2\sigma_i^2}, \tag{6}$$

where

$$\sum_{i=1}^{M} c_i = 1, \tag{7}$$

with $c_i$ representing a weight factor and $M$ being the number of distributions.

In SimPatico, this model is implemented by the mixture of two gaussian functions ($M = 2$). The variance of the first one accounts for the contribution of a purely gaussian noise, where $\sigma_1$ is the variance of the first gaussian function, $\sigma_{total}$ is the variance of the distribution, and $c_1$ and $c_2$ are the weights of both gaussian distributions. The input parameter $\rho$ describes the noise impulsiveness and it is configurable in the simulator. The relationship between the variances is given by

$$\sigma_1 = \frac{\sigma_{total}}{c_1 + \rho c_2} \tag{8}$$

where $\sigma_2 = \rho\sigma_1$.

### 4.3. Symmetric $\alpha$-Stable Model

Many applications must consider non-gaussian models for realistic approaches since gaussian processes are not appropriate for modelling impulsive noises [22, 23]. Therefore, several works suggest the $\alpha$-stable model as a solution for the non-gaussian impulsive noise model. The $\alpha$-stable model has demonstrated to be a better fit to real data in impulsive noise conditions if compared to the gaussian model [23]. Thus, SimPatico implements the Symmetric $\alpha$-Stable (S$\alpha$S) model, from the $\alpha$-stable class, because it has proved to be very useful in modeling impulsive noise [24].

As an univariate stable distribution, this model is characterized by four parameters [25]: the index of stability $\alpha$, which can vary from 0 to 2; the scale parameter $\sigma$; the skewness parameter $\beta$, which can assume values from -1 to 1; and the shift parameter $\mu$. It is important to note that when $\alpha = 2$, the stable distribution is gaussian, so $\sigma$ is the standard deviation, $\beta$ can be taken as zero and $\mu$ is the mean. When $\alpha$ is close to 1, the model assumes a Cauchy distribution.

An $\alpha$-stable distribution is described by its characteristic function (for $\alpha \neq 1$):

$$\begin{aligned} E(\alpha,\beta,\sigma,\mu,\theta) = \\ \exp\Big\{ -\sigma^\alpha|\theta|^\alpha(1 - j\beta(\mathrm{sign}(\theta)\tan\frac{\pi\alpha}{2}) + j\mu\theta)\Big\}. \end{aligned} \tag{9}$$

If $\alpha = 1$, the characteristic function assumes the following form:

$$E(\alpha,\beta,\sigma,\mu,\theta) = \\ \exp\Big\{ -\sigma|\theta|(1 + j\beta\frac{2}{\pi}(\text{sign}(\theta)\ln|\theta|) + j\mu\theta)\Big\}. \quad (10)$$

where $\text{sign}(\theta)$ is given by:

$$\text{sign}(\theta) = \begin{cases} 1 & \text{if} \quad \theta > 0, \\ 0 & \text{if} \quad \theta = 0, \\ -1 & \text{if} \quad \theta < 0. \end{cases} \quad (11)$$

In the case of S$\alpha$S, the model imposes $\beta = 0$ and $\mu = 0$. The parameters $\alpha$ and $\sigma$ are configurable by the user to determine a specific GSNR and impulsiveness degree. Those parameters select the severeness of the noise in the simulator. Thus, its characteristic functions is reduced to:

$$E(\alpha,\sigma,\theta) = \exp\Big\{ -\sigma^{\alpha}|\theta|^{\alpha}\Big\}. \quad (12)$$

### 4.4. Reverberation Model

Lastly, the reverberation model follows the plate-class topology [26] and allows for custom values of pre-delay and sample rate. The process of reverberation consists of a number of steps described in Figure 2.

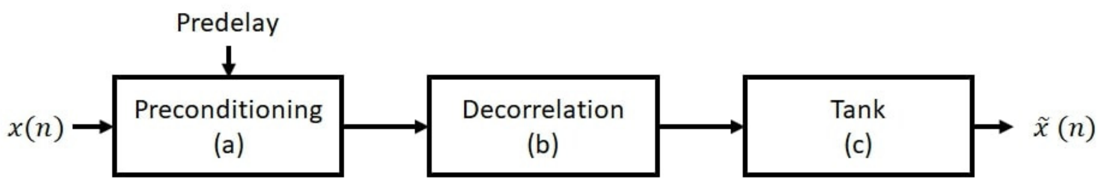



**Figure 2:** Flowchart of reverberation model.

Such steps are:

1. Preconditioning the input by applying a custom pre-delay on it and following a single-pole low-pass filter of $-3$ dB cutoff frequency;
2. Decorrelating by applying four series all-pass filters;
3. Feeding into a tank processing block to simulate the amplitude decay of the reverberating signal.

In the decorrelation process, the all-pass filters are given in $z$ domain by

$$H_1(z) = \frac{C + z^{-k_i}}{1 + Cz^{-k_i}}, \quad (13)$$

where the constants are $C = 0.5$, called diffusion constant, and $k_1 = 142$, $k_2 = 107$, $k_3 = 379$, and $k_4 = 277$, for the $i$th filter respectively.

In the tank process, the signal passes by three filters in cascade. The first one is modulated all-pass filter, given by

$$H_2(z) = \frac{-C + z^{-k}}{1 - Cz^{-k}}, \tag{14}$$

where $C$ is the diffusion constant, and $k$ is specified in order to obtain an amplitude of $(8/29761) \cdot f_s$ for 1 Hz, which $f_s$ is the sample rate.

Then, the signal is delayed again by a low-pass filter, given by

$$H_3(z) = \frac{1 - \phi}{1 - \phi z^{-1}}, \tag{15}$$

where $\phi$ is the high-frequency damping, proportional to the attenuation of high frequencies in the reverberation output, and assumes the value $\phi = 5 \times 10^{-4}$.

The last filter is an all-pass filter representing the decay factor given by

$$H_4(z) = \frac{C + z^{-k}}{1 + Cz^{-k}}, \tag{16}$$

where $C$ is the diffusion constant.

Finally, the signal $\tilde{x}(n)$ generated is used for further simulation purposes regarding reverberation.

# 5. SimPatico Simulator

SimPatico contains a variety of TDOA methods, including angle of arrival (AoA) and robust algorithms that are not scope of this work. As previously mentioned, there are also degradation models for gaussian and non-gaussian channels such as GMM and S$\alpha$S processes. Results are produced in terms of performance metrics of the direction of arrival (DoA) algorithms and they are saved in data files and plots. The simulator is available in Gihub at https://github.com/vicentesousa/SimPatico[1]. The flow of code execution and its most relevant functionalities are presented in Figure 3 and described in the following sections.

## 5.1. Execution flow

Initially, the simulation campaigns are composed of parameter files, which contain parameters about the source signal, number of sensors, the distance between them, and other relevant information about the simulation. Also, one can set the noise and channel model parameters optionally, separating the results according to all model combinations. In the checking data procedure, all parameters are passed into a structure to make the code reusable and maintainable. The previously mentioned parameters are analyzed to ensure the consistency of them, emitting an error message if necessary.

[1]The simulator will be made publicly available after confirmation of publication of this article.

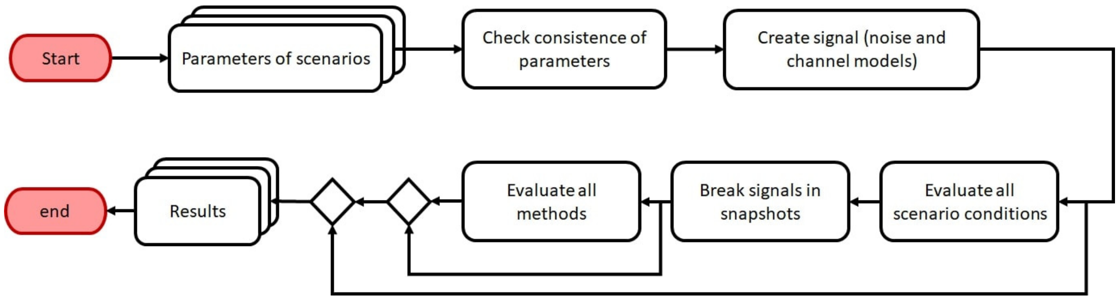



**Figure 3:** Flowchart of campaigns of the simulator.

In the creating signal step, the signal is created based on the type of the source signal, such as sines, Zadoff-Chu sequence, random, and deterministic audio signals. In the case of a sine-based signal, the signal frequency is required; otherwise, it is not necessary. Those signals are available in real or complex forms and the distortion models can be added to them. SimPatico is also prepared to process audio files from field measurements. In the sequence, the simulation is executed in snapshots. There, the angle of arrival is estimated through the chosen TDOA methods and figures of merit are also calculated for further results. The execution of snapshots is then held for all ranges of SNR, number of iterations and snapshots, every selected source signal and so on.

The results of each simulation in terms of performance metrics (e.g., RMSE, absolute error, probability of resolution) are stored in a structure, later saved in an output file. Lastly, all the performance metrics are plotted and saved for further analysis. The paths for the output file and plots are automatically generated. Implementation and organization details of the SimPatico are described in terms of folders and function codes to allow the reuse of the simulator.

In the `simulations` folder, the user can specify a list of simulations by choosing parameters and scenarios to build a campaign. The simulation campaign produces all results in the `results` folder with all data files and plots. The performance metrics are provided based on a dependent variable that can be specified by the user, such as SNR, GSNR, number of sensors and length of source signal.

## 6. Use Cases

In this section, we illustrate various examples of performance evaluation studies to show the utility of the SimPatico for the analysis of TDOA methods. Cases where gaussian and non-gaussian distortions are presented in Section 6.1, 6.2 and 6.3, followed by the analysis of resolution and the number of sensors in Sections 6.4 and 6.5. Finally, Section 6.6 presents a reverberation analysis. Such results can be viewed as a basis for comparison and validation of TDOA algorithms available in the literature.

We compare the performance of all mentioned TDOA methods for a given scenario. As for the setting for all the analysis that are conducted in this section, a “gong” sound is produced, with two sensors 0.2427 meters away from each other and the sampling frequency is 8192 Hz. The

SNR is set from -20 up to 40, and 1000 Monte Carlo runs are performed, with 1000 samples taken in each simulation run. All source signals are not affected by reverberation and we analyze the effects of different noises over the performance of the TDOA methods.

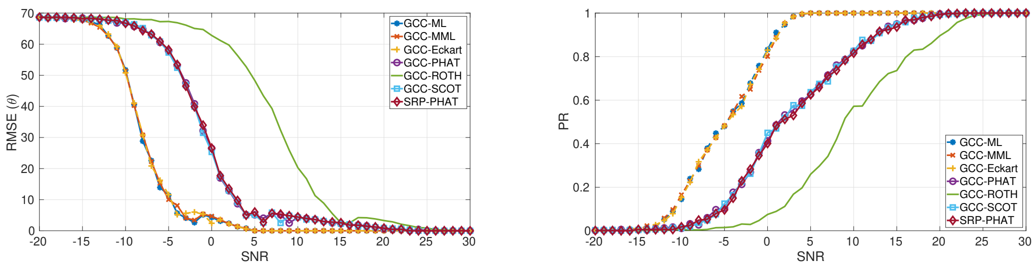
The angle estimation accuracy is assessed by the root-mean-square error (RMSE) in degrees and the probability of resolution (PR) versus the signal-to-noise ratio (SNR). However, in the case of non-gaussian noise model, the undefined variance prevents the SNR to be used as a measurement of signal quality. Thus, we use the geometric signal-to-noise ratio (GSNR) [27] instead of the SNR in the case of S$\alpha$S processes and the weighted variance (Equation 8) to compute SNR in the case of GMM. The GSNR is given by

$$\mathrm{GSNR} = \frac{1}{2C_g}\left(\frac{A}{S_0}\right)^2, \tag{17}$$

where the normalization constant $C_g = e^{C_e} \approx 1.78$ is the exponential of the Euler constant ($C_e$), used to ensure that GSNR corresponds to SNR when the channel is Gaussian, $S_0$ is the geometric power of a S$\alpha$S random variable; and $A$ is the root-mean-square value of the signal.

### 6.1. Gaussian noise analysis

Firstly, in Figure 4, we analyze one of the most common scenarios in communication systems, where the signals are affected by the gaussian noise.




(a) RMSE for source signal affected by gaussian noise.

(b) Probability of resolution for source signal affected by gaussian noise.

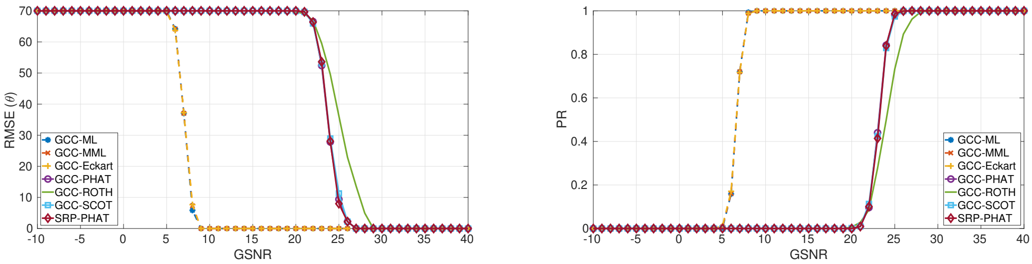
**Figure 4:** Simulation results for source signal affected by gaussian noise.

In Figure 4a, we obtain the RMSE versus SNR for all TDOA methods. As expected, the Eckart, the ML and the MML filters are superior than the others. In those conditions, the ML has optimal solution achieving the lowest RMSE. Also, the Eckart and MML reach similar performance as more complex computational solutions are given by prior information used by them. The classical filters obtain expected performance by reaching high PR only at high SNR values, as presented in Figure 4b.

The ROTH filter obtains the highest error and the lowest PR for all SNR due to its design focused on random signals with high SNR. Considering that the signal is cyclic, the GCC-ROTH only reaches an adequate performance in high values of SNR.

## 6.2. GMM analysis

A second simulation scenario models an impulsive noise present in the signal by the GMM, a non-gaussian model. In the GMM analysis, we observe a major performance difference among the methods, in performance, as illustrated in Figure 5.




(a) RMSE for source signal affected by gaussian mixture modelled noise.

(b) Probability of resolution for source signal affected by gaussian mixture modelled noise.

**Figure 5:** Simulation results for source signal affected by gaussian mixture modelled noise.

Figure 5a presents the RMSE versus the average SNR, showing that the methods with prior information from the noise have similar performance, while the other methods show a higher RMSE even for high SNR values. In Figure 5b, the PR versus average SNR is presented, demonstrating the infeasibility of methods such as GCC-ROTH, GCC-SCOT, GCC-PHAT, and SRP-PHAT due to non-detection in high SNR values. As expected, the GCC-ROTH is the worst case since it is designed for high SNR values.

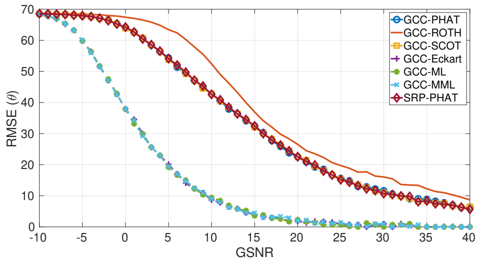
We notice that in this scenario, with $\rho = 1000$, representing a highly impulsive channel, the methods based on Eckart, ML, and MML weighting functions reach high PR values only above 10 dB of SNR approximately. They would be a reasonable choice in this case given that none of them is designed as a robust method for impulsiveness scenarios.

## 6.3. S$\alpha$S analysis

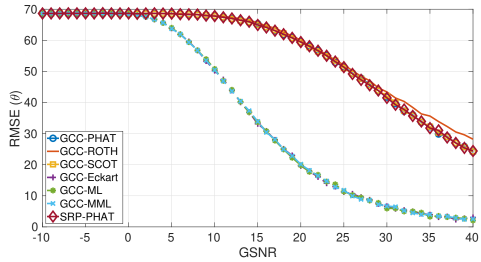
Figure 6 presents the performance metrics of the methods for impulsive noise based on S$\alpha$S modeling. First, we analyze low impulsiveness noise with $\alpha = 1.9$, represented in Figures 6a and 6c. Then, we analyze for higher impulsiveness noise condition with $\alpha = 1.7$, shown in Figures 6b and 6d. Both cases represent a model for impulsive noise in terms of GSNR and the degree of impulsiveness given by $\alpha$ parameter.

In Figures 6a and 6c, although the noise has still some similarity to a gaussian noise, the SNR requirements for a appreciable probability of resolution and reduction of RMSE are higher in terms of GSNR. In Figures 6b and 6d, the considerable impact of the impulsive noise in performance of the algorithms is even clearer.

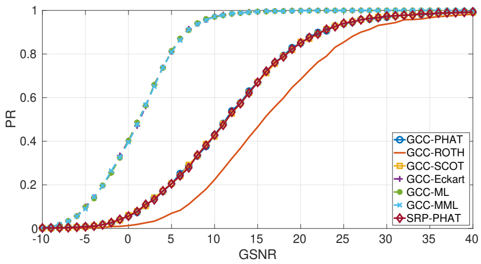
A remarkable result seen in those results are the overall efficacy fall of all filters when $\alpha$ is decreased. The already mentioned filters with the superior effectiveness needed a much higher SNR value in order to achieve an appropriate probability of resolution. The remaining required a SNR higher than 40 dB for the same result, in many applications, is not practical.



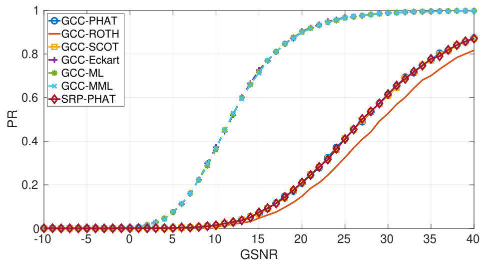

(a) RMSE for source signal affected by $\alpha$-stable modelled noise ($\alpha = 1.7$).




(b) RMSE for source signal affected by $\alpha$-stable modelled noise ($\alpha = 1.3$).




(c) Probability of resolution for source signal affected by $\alpha$-stable modelled noise ($\alpha = 1.7$).




(d) Probability of resolution for source signal $\alpha$-stable modelled noise ($\alpha = 1.3$).

**Figure 6:** Simulation results for source signal affected by $\alpha$-stable modelled noise.

## 6.4. Resolution analysis

We also assess the resolution of the methods by varying the length of the input signals. This evaluation exhibits how fast a method decreases its performance with less samples. We evaluate the performance of the methods, looking for RMSE versus the length of samples ($N$) when SNR = 3dB (gaussian noise), as shown in Figure 7.

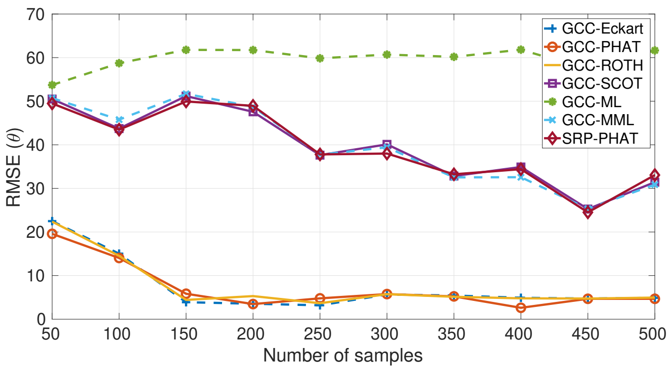



**Figure 7:** Resolution of the TDOA methods.

In this case, we can clearly notice the difference of performance among the techniques. The general tendency is to decrease the error with more number of samples. In the case of Eckart, PHAT and ROTH filters, the RMSE is close to zero with more than 150 samples.

## 6.5. Number of sensors analysis

Differently from direction of arrival techniques in which the resolution is given by the number of sensors, in TDOA methods, the resolution is given by the length of the snapshot. However, they can increase their performance using more number of sensors to estimate the delay.

For example, Figure 8 presents the RMSE versus SNR of SRP-PHAT from 2 up to 8 sensors. In the case of SRP-PHAT method, the performance tends to increase when more sensors are used, reaching similar RMSE values of GCC with Eckart, ML, and MML filters when 8 sensors are employed.

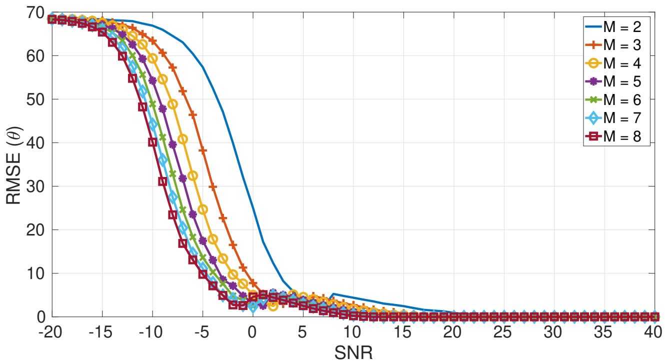



**Figure 8:** RMSE versus SNR of SRP-PHAT in number of sensors analysis.

## 6.6. Reverberation analysis

We evaluate the reverberation based on the pre-delay parameter in the reverberation channel model. In Figure 9 a scenario with gaussian noise and reverberation with delay $D = 0.1$ is shown. We compare GCC-PHAT and SRP-PHAT in terms of RMSE.

Clearly, the GCC-PHAT does not reach an adequate performance for all SNR values, while the SRP-PHAT achieves null error at approximately 0 dB of SNR. As expected, the SRP-PHAT is an appropriate method for reverberation conditions.

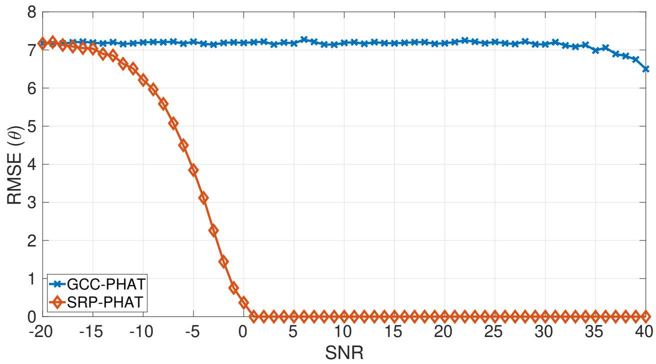



**Figure 9:** RMSE versus SNR in reverberation analysis.

## 7. Conclusions

Estimating the TDOA for different scenarios is crucial to a wide variety of localization systems. There are many techniques for such evaluation, each one with a different performance trade-off regarding the parameters and environment conditions. Thus, the access of a unified platform to analyze the add-value of existing or new strategies is a beneficial contribution to those interested in these assessments. Simulators which perform TDOA calculations for localization purposes are available, but they are limited in some aspect, since they are focused on indoor localization. SimPatico was then developed based on the needs for evaluation of TDOA methods in different settings, accounting for various source signals, noise assumptions and the limitations of previously released tools.

One can estimate the TDOA by calculating the cross-correlation between two signals and extract the position of its maximum value. However, this operation may not return the desired result due to aspects such as noise. To account for this issue, some methods were developed, and they can be separated into two categories: generalized cross-correlation and steered-response power phase transform. The first one can be seen as the application of filters for the cross-correlation operation, each one with different properties; the latter is a realization of the beamforming process, where the number and position of each sensor plays a determinant role on the overall performance of the system.

SimPatico contains many variations for the noise model, which is eventually added to the source signal. These include gaussian and non-gaussian functions. For the latter, the simulator possesses implementations of gaussian mixtures and $\alpha$-stable models. Also, it is possible to apply reverberation, so that a more realistic analysis can be conducted.

The main functions and the structure of the simulator are explained, beginning from the signal creation to where the outputs are generated and stored. Lastly, some generated results are shown as examples. One can observe that the ML filters have superior performance for severe noise conditions, as expected. Furthermore, the presence of non-gaussian noise components on the source signal deteriorates the overall efficiency of the implemented methods, although the same filters possess the best behavior. Thus, it is necessary to further conceive strategies that produce a better performance than the ones that were already used. We claim that SimPatico could help the scientific community in this task.

## 8. Acknowledgments

This work was partially supported by Samsung Eletronica da Amazonia Ltda., under the auspice of the informatic law no 8.387/91.

## References

[1] Q. Liu, R. Liu, Z. Wang, Y. Zhang, Simulation and Analysis of Device Positioning in 5G Ultra-Dense Network, in: IWCMC, 2019.

[2] A. Kakkavas, M. H. C. Garcia, R. A. Stirling-Gallacher, J. A. Nossek, Multi-Array 5G V2V Relative Positioning: Performance Bounds, in: GLOBECOM, 2018.

[3] Y. Guo, X. Wang, C. Wu, Q. Fu, N. Ma, G. J. Brown, A Robust Dual-Microphone Speech Source Localization Algorithm for Reverberant Environments, in: INTERSPEECH, volume 08-12-Sept, 2016.

[4] R. Liang, N. Peng, L. Zhou, X. Meng, Y. Hu, Y. Shen, X. Xue, Fault Location Method in Power Network by Applying Accurate Information of Arrival Time Differences of Modal Traveling Waves, IEEE Transactions on Industrial Informatics 16 (2020) 3124–3132.

[5] J. Sun, J. Wen, Target location method for pipeline pre-warning system based on HHT and time difference of arrival, Measurement 46 (2013).

[6] N. Moayeri, M. O. Ergin, F. Lemic, V. Handziski, A. Wolisz, PerfLoc (Part 1): An extensive data repository for development of smartphone indoor localization apps, in: PIMRC, IEEE, 2016.

[7] R. Montoliu, E. Sansano, J. Torres-Sospedra, O. Belmonte, IndoorLoc platform: A public repository for comparing and evaluating indoor positioning systems, in: IPIN, IEEE, 2017, pp. 1–8.

[8] T. Jankowski, M. Nikodem, SMILe - Simulator for Methods of Indoor Localization, in: IPIN, IEEE, 2018, pp. 206–212.

[9] S. He, X. Dong, High-accuracy localization platform using asynchronous time difference of arrival technology, IEEE Transactions on Instrumentation and Measurement 66 (2017) 1728–1742.

[10] R.-R. e. a. Wang, Design of a tdoa location engine and development of a location system based on chirp spread spectrum, SpringerPlus 5 (2016).

[11] M. Brandstein, D. Ward, Microphone Arrays, Digital Signal Processing, Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[12] C. Knapp, G. Carter, The generalized correlation method for estimation of time delay, IEEE Transactions on Acoustics, Speech, and Signal Processing 24 (1976) 320–327.

[13] P. R. Roth, Effective measurements using digital signal analysis, IEEE Spectrum 8 (1971) 62–70.

[14] G. Carter, A. Nuttall, P. Cable, The smoothed coherence transform, Proceedings of the IEEE 61 (1973) 1497–1498.

[15] G. C. Carter, A. H. Nuttall, P. G. Cable, The smoothed coherence transform, Proceedings of the IEEE 61 (1973) 1497–1498.

[16] C. Eckart, Optimal Rectifier Systems for the Detection of Steady Signals, Technical Report, UC San Diego: Library – Scripps Digital Collection, 1952.

[17] E. J. HANNAN, P. J. THOMSON, Estimating group delay, Biometrika 60 (1973) 241–253.

[18] M. S. Hosseini, A. Rezaie, Y. Zanjireh, Time difference of arrival estimation of sound source using Cross Correlation and modified maximum likelihood weighting function, Scientia Iranica 24 (2017) 0–0.

[19] J. Chen, J. Benesty, Y. Huang, Time Delay Estimation in Room Acoustic Environments: An Overview, EURASIP Journal on Advances in Signal Processing 2006 (2006) 026503.

[20] J. H. DiBiase, H. F. Silverman, M. S. Brandstein, Robust Localization in Reverberant Rooms, 2001, pp. 157–180.

[21] W. J. Zeng, H. C. So, L. Huang, lp-MUSIC: Robust direction-of-arrival estimator for impulsive noise environments, IEEE Transactions on Signal Processing 61 (2013) 4296–4308. doi:10.1109/TSP.2013.2263502.

[22] P. Borghesani, J. Antoni, Cs2 analysis in presence of non-gaussian background noise - effect on traditional estimators and resilience of log-envelope indicators, 2017.

[23] D. Pena, C. Lima, M. Dória, L. Pena, A. Martins, V. Sousa, Acoustic Impulsive Noise Based on Non-Gaussian Models: An Experimental Evaluation, Sensors 19 (2019) 2827.

[24] C. L. Nikias, M. Shao, Signal Processing with Alpha-stable Distributions and Applications, Wiley-Interscience, New York, NY, USA, 1995.

[25] G. Samoradnitsky, M. Taqqu, Non-Gaussian Random Processes: Stochastic Models with Infinite Variance, 1 ed., Chapman and Hall/CRC, 1994.

[26] J. C. Dattorro, Effect Design: Part 1: Reverberator and Other Filters, Technical Report, 1997.

[27] J. G. Gonzalez, J. L. Paredes, G. R. Arce, Zero-order statistics: A mathematical framework for the processing and characterization of very impulsive signals, IEEE Transactions on Signal Processing 54 (2006).