

## RESEARCH ARTICLE

## Estimating speaker direction on a humanoid robot with binaural acoustic signals

Pranav Barot<sup>1\*</sup>, Katja Mombaur<sup>1,2</sup>, Ewen N. MacDonald<sup>1</sup>

**1** Department of Systems Design Engineering, University of Waterloo, Waterloo, Ontario, Canada, **2** Karlsruhe Institute of Technology (KIT), Institute of Anthropomatics and Robotics (IAR), Optimization and Biomechanics for Human-Centred Robotics, Karlsruhe, Germany

\* [pbarot@uwaterloo.ca](mailto:pbarot@uwaterloo.ca)

## Abstract

To achieve human-like behaviour during speech interactions, it is necessary for a humanoid robot to estimate the location of a human talker. Here, we present a method to optimize the parameters used for the direction of arrival (DOA) estimation, while also considering real-time applications for human-robot interaction scenarios. This method is applied to binaural sound source localization framework on a humanoid robotic head. Real data is collected and annotated for this work. Optimizations are performed via a brute force method and a Bayesian model based method, results are validated and discussed, and effects on latency for real-time use are also explored.

## OPEN ACCESS

**Citation:** Barot P, Mombaur K, MacDonald EN (2024) Estimating speaker direction on a humanoid robot with binaural acoustic signals. PLoS ONE 19(1): e0296452. <https://doi.org/10.1371/journal.pone.0296452>

**Editor:** Ying Shen, Tongji University, CHINA

**Received:** August 12, 2023

**Accepted:** December 11, 2023

**Published:** January 2, 2024

**Copyright:** © 2024 Barot et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Data Availability Statement:** All recordings and ground truth data are found at this link: <https://doi.org/10.5281/zenodo.8172587>.

**Funding:** E.M - Natural Sciences and Engineering Research Council of Canada, RGPIN-2021-03085 The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Competing interests:** The authors have declared that no competing interests exist.

## 1 Introduction

Speech is one of the most important forms of human communication and a key element of social interaction. Thus, to better integrate humanoid robots into society and augment human-robot interaction, it is important for them to achieve speech interactions that are similar to human-human interactions. Speech interactions are a complex phenomenon that includes both verbal and non-verbal behaviour. One aspect of this non-verbal behaviour is how talkers and listeners orient their head and body relative to their conversational partner.

In the present study, we focus here on a sub-task of identifying the direction of arrival (DOA) of human speech. This information is necessary for humanoid robots to interact with humans in realistic and natural ways, such as orienting to and tracking human conversational partners (who may move during the conversations), or handling interactions that involve multiple conversational partners.

Much work has been done on sound source localization (SSL) by robots (for a review see [1]) and many of the methods are based on cues that are used by humans to localize sound sources. Given an array of two or more microphones that are spatially separated, the sound from a source will arrive at each microphone at different times. Thus, by measuring the time difference of arrival between microphones, and knowing the geometry of the microphone array, it is possible to estimate the DOA of the source. This method is analogous to the use of inter-aural timing (ITD) difference cues used by humans. A related approach involves the use of beamforming. The output level of a beamformer should be higher if it is steered in the direction of the source. Thus, DOAs can be estimated by finding look directions which correspond to maxima of the

beamformer output levels. If an object is present between the microphones in an array, that object will alter the acoustic field and can vary the level of the signals received at the different microphones. For example, if the object is large compared to the wavelength of the source, the object can cast an acoustic “shadow”. Thus, microphones where the object is located in the direct path to the source will record lower levels than those where the object is not in the path. This is analogous to inter-aural intensity differences (IID) used by humans (where the head can result in substantial level differences between the ears at high frequencies). For different DOAs, the geometry of the irregularly-shaped human pinnae (the part of the ear that is on the head) results in patterns of constructive and destructive interference that will vary with DOA. These spectral notches “colour” the sound received by the ear. Thus, by estimating the patterns of spectral notches, it is possible to infer the DOA. Given the complexity of these patterns and the relationship with DOA, this use of this spectral approach relies on learning methods.

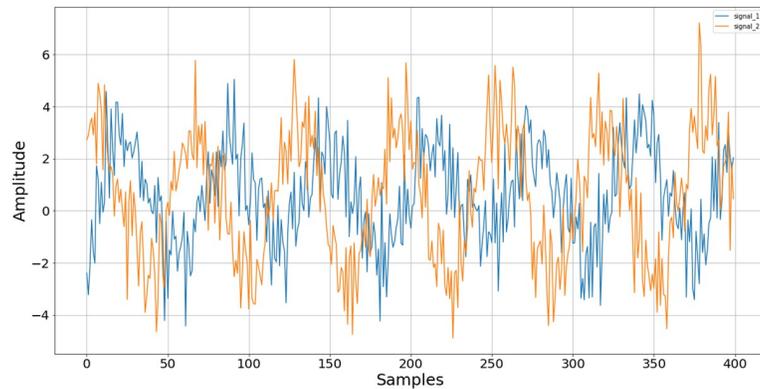
A further factor to consider in SSL is the effect of the environment. In general, sound sources radiate sounds in multiple directions. Surfaces that are present in the environment (e.g., walls, floor, ceiling, furniture, etc.) will reflect a portion of the incident sound. Thus, the sound signal recorded at a microphone will be a sum of the acoustic signal from the direct path between the source and the microphone and all the other paths that involve one or more reflections. In the context of DOA estimation, the paths that involve reflection will have a different DOA than that of the direct path.

In the context of human speech interactions, another key factor is the timing of turns. Previous work investigating human conversation has found that talkers start their turn approximately 200-300 ms after their partner has finished their turn [2–4]. To achieve human-like interactions, it is necessary for a humanoid robot to respond within a similar time frame. The latency of generating DOA estimates will limit how quickly a humanoid robot can respond to movement of a current talker or orient towards a new talker. Works such as [5, 6] consider accurate DOA estimation on robotic systems, but also require a consideration for latency and turn-taking in the context of human-robot conversational scenarios.

Our work evaluates and optimizes a pipeline consisting of two main stages. The first stage continuously generates DOA estimates based on the acoustic signals received from two microphones placed on the head of a humanoid robot. The second stage categorizes these DOA estimates as being “good”, that is the estimate likely corresponds with the direct signal from a human talker rather than background noise or a reverberant echo. Using a manually collected and labeled dataset, we investigate the performance of the pipeline’s ability to detect direct human speech among background noise and self-generated robot sounds, accurately estimate the direction of arrival, and account for latency of detection. The unique parameters of the pipeline are optimized via either a brute force approach or a more efficient and useful Bayesian optimization approach, which sheds light on how the pipeline’s performance depends on each chosen parameter.

## 1.1 DOA estimation

The first main stage in the pipeline is to generate DOA estimates based on the acoustic signals received at multiple microphones. In the present study we consider the case where there are two spatially separated microphones. For this case, the simplest approach to estimate direction of arrival is to examine the cross-correlation of signals from the two microphones to estimate the difference in arrival time between the two microphones. These received signals can be streamed in real-time, or can be processed after being recorded. The estimate of difference in arrival time can then be resolved to a direction given that the geometric setup of the microphones is known.



**Fig 1. Example of two simulated microphone signals.** Visualization of the signals  $y_1$  (blue) and  $y_2$  (orange).

<https://doi.org/10.1371/journal.pone.0296452.g001>

**1.1.1 Cross-correlation and beamforming.** Beamforming is a method used to improve the directionality of an array of receivers. A simple method is a delay-and-sum technique, where the signals from each receiver are delayed by a fixed amount that varies across receivers and are then summed together. As noted earlier, one can estimate a DOA using a beamformer by steering the beam across all angles and finding the direction that results in the largest signal. When the array consists of only two microphones, the delay-and-sum beamforming technique is closely related to the cross-correlation based DOA method to estimate the maximal time alignment/beam direction.

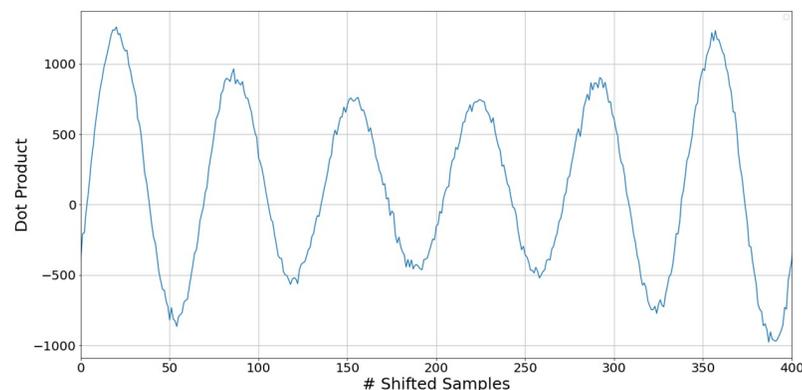
Consider two waves measured at receiver 1 and receiver 2, as per the equations below, with some added Gaussian noise.

$$\begin{aligned} y_1 &= 2\sin(x - 1) + \mathcal{N}(0.5, 1.2) \\ y_2 &= 3\cos(x - 0.5) + \mathcal{N}(0.5, 1.2) \end{aligned} \quad (1)$$

Their raw measured amplitude over time appears as in Fig 1.

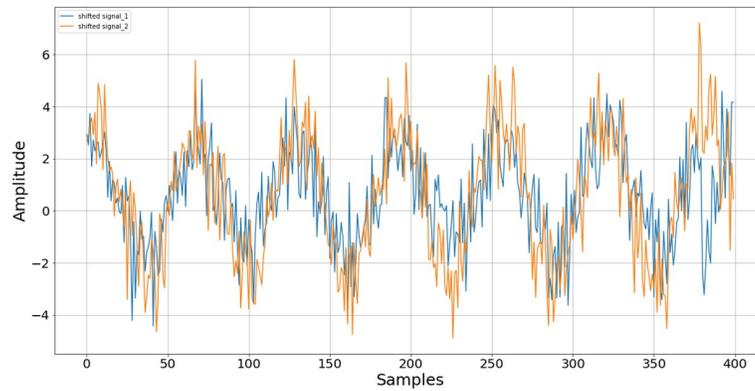
Applying a time domain cross-correlation operation directly results in an output as in Fig 2.

The maximum value appears at  $n = 20$  samples, indicating that this value best aligns the two received signals. After shifting one signal by the required 20 samples, the resultant is now as in Fig 3. Evidently, the signals are well aligned after using the estimate from the cross-correlator.



**Fig 2. Plot of the cross correlation of  $y_1$  and  $y_2$  for positive lags.**

<https://doi.org/10.1371/journal.pone.0296452.g002>



**Fig 3. Visualization of the signals  $y_1$  (blue) and  $y_2$  (orange) after applying a delay of 20 samples to  $y_1$ .**

<https://doi.org/10.1371/journal.pone.0296452.g003>

**1.1.2 Variations on cross correlation.** Since traditional cross-correlators are computationally expensive and sensitive to background noise and reverberation, spectral domain methods are used in this work. Interaural timing differences are estimated using the Wiener-Khinchin relation for the cross-power spectrum, using the Fourier transforms of two recorded signals  $x$  and  $y$ .

$$G_{xy} = X[f]Y[f]^* \tag{2}$$

This relation is used to estimate cross-correlation output of  $x$  and  $y$  as per the following generalized formulation, the *argmax* of which indicates the ITD between the two microphones [7].

$$\hat{R}_{xy} = \int_{-\infty}^{\infty} \psi(f)G_{xy}(f)e^{i2\pi f\tau} df \tag{3}$$

The cross-correlation vector is then the inverse Fourier transform of this result.

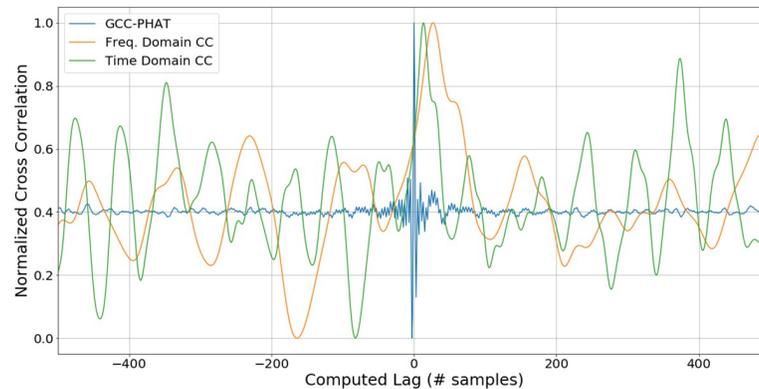
**1.1.3 Spectral domain cross correlation.** The spectral domain cross correlation comes with no whitening transform on the cross correlator. This results in the general estimator as in Eq (4). The advantage is the computational efficiency of not requiring a delay-and-sum operation in the time domain while still generating an estimate of the cross-correlation output.

$$\psi_{CC}(f) = 1 \tag{4}$$

**1.1.4 Generalized cross correlation—Phase transform.** The phase transform (GCC-PHAT) pre-whitens the cross-correlation response using the value of  $\psi$  as in Eq (5), providing robustness against reflections in difficult auditory environments [1].

$$\psi_{PHAT}[f] = \frac{1}{|G_{xy}(f)|} \tag{5}$$

**1.1.5 Generalized cross correlation—Smoothed coherence transform.** The smoothed coherence transform (SCOT) aims to reduce the error contributed by both signals  $X$  and  $Y$ , where the PHAT may not be able to adequately handle the case where  $G_{xx} \sim 0$  or  $G_{yy} \sim 0$  in



**Fig 4. Cross correlation with different estimators.** The output of three cross correlation estimators, GCC-PHAT (blue), frequency domain (orange), and time domain (green) are plotted for a 350 ms frame of speech.

<https://doi.org/10.1371/journal.pone.0296452.g004>

lower frequency bands. This provides the SCOT pre-whitening, as in Eq (6).

$$\psi_{scot}[f] = \frac{1}{\sqrt{G_{xx}(f)G_{yy}(f)}} \quad (6)$$

These cross-correlation methods are visualized by their output on a frame of 350ms containing speech. Fig 4 shows the results from a time domain cross correlation, a frequency domain cross correlation, and the GCC-PHAT.

The two naive cross-correlation methods generate noisier outputs, as their local maxima are quite similar to the global maxima. This is attributed to reflections and reverberation present in the audio frame, which make this problem more complex. However, the GCC-PHAT is able to find one peak that is far more prominent than the rest, as a result of the applied pre-whitening transform. The prominence of the peak increases the confidence that the estimated timing difference is in fact due to the direct speech, and not a stray reflection or reverberation.

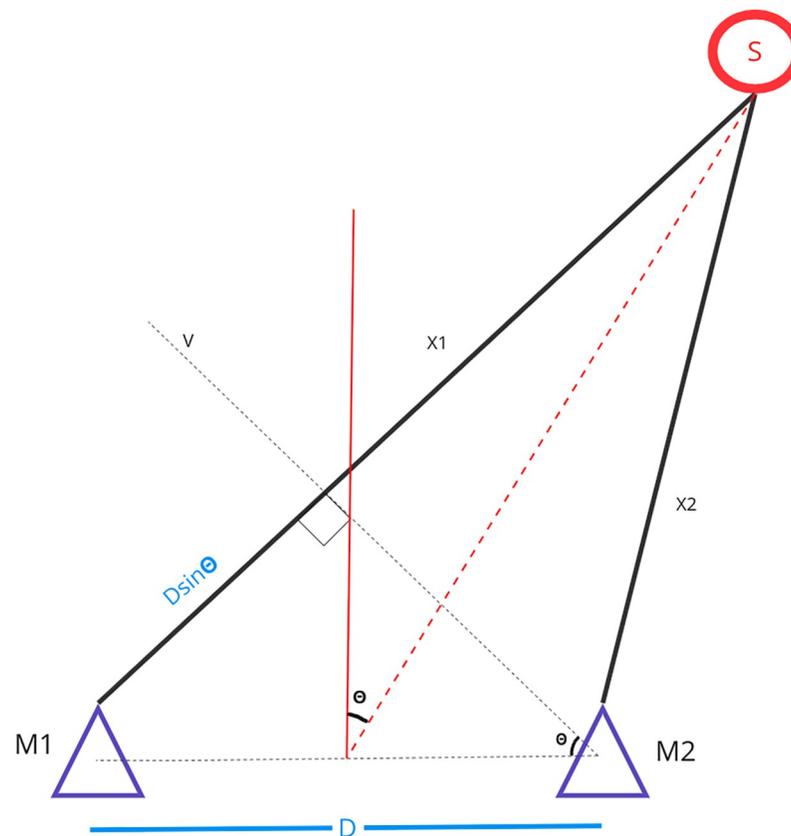
The performance of these methods will depend heavily on the chosen audio frame size, background noise and the environment of the robot. Optimization performed in later sections will indicate which method works best for the given tasks.

## 1.2 Generating the direction of arrival

Once the timing difference has been determined, a geometric model is used to estimate the direction of arrival of the sound source. A simplified description is presented in Fig 5, showing two microphones M1 and M2, separated by a distance  $D$ , with two unique path lengths  $X1$  and  $X2$  to a sound source  $S$ . The sound source  $S$  is assumed to be sufficiently far away from the microphone array, such that the transmitted sound waves are planar, and can be processed with the right triangle generated by M1, M2 and the incident sound wave.

Given the right triangle made by M1, M2 and the path length  $X1$ , with an angle of  $\theta$ , the opposite then becomes  $D\sin\theta$ , given the distance  $D$  between the two microphones. This distance represents the extra distance the wavefront must travel to reach M1 once it has reached M2. This distance is directly computed from the timing difference  $\tau$ , and so the measured quantities are related as in Eq (7).

$$D\sin\theta = \tau v_{sound} \quad (7)$$



**Fig 5. Geometry used to estimate DOA from two microphones (M1, and M2) that are separated by length D.**

<https://doi.org/10.1371/journal.pone.0296452.g005>

Since the geometry of the robot head setup is not exactly as in this simplified model, the timing difference is used in the Woodworth-Schlosberg model [8] to estimate the DOA on a spherical robotic head, such as the REEM-C's. Eq (8) shows the modification that now maps the timing differences  $\tau$  as a function of the DOA. This new model accounts for the extra radial distance the wavefront must travel to reach the microphone on the other side of the head. This mapping is used to find the corresponding value of the direction of arrival  $\theta$  in real-time. The ear-to-ear distance  $D$  of the REEM-C is calibrated by measuring the ITDs at a number of known angles, and calculating the distance that would result in these measurements. With this method, the average ear-to-ear distance is computed as  $d = 0.255\text{m}$ .

$$\tau(\theta) = \frac{D}{2 * v_{\text{sound}}} (\theta + \sin(\theta)) \quad (8)$$

### 1.3 Environmental considerations

In a real-world application, it is likely that a humanoid robot will interact with humans in an environment that has some background noise. Voice activity detection (VAD) is a common problem in audio processing contexts, where the goal is to identify when speech is present in an audio recording. When computing the DOA on the REEM-C, the streamed audio will contain a variety of sounds that may not be speech, such as robot operation noises and ambient

noise. Reverberant echoes from a talker will also generate spurious DOA estimates as the direction of these echoes is not the same as that of the direct path from the source. We explore two methods to classify if a DOA estimate is “good” (i.e., the estimate is likely to correspond with the direct path of speech from a talker).

**1.3.1 Power onsets.** If the microphone signal is split into frames with some window length, the energy in each frame will vary over time based on the fluctuations from the sound source. In a reverberant environment, when a talker stops speaking, it will take some time for the sound energy to decay. When a talker begins speaking, the sound from the direct path will arrive at the microphone before later reflections. Thus, a frame that has significantly more energy than the previous frame (i.e., an onset) is likely to have more direct path energy than in the reverse case. Here we choose to use successive frame power ratios rather than differences, where an onset is detected if the power ratio exceeds a certain threshold. For a certain frame  $F_i$ ,

$$F_i = \begin{cases} \text{speech frame} & \text{if } \delta_{high} > \frac{1}{N} \sum_j \frac{F_{i,j}^2}{F_{i-1,j}^2} > \delta_{low} \\ \text{non-speech frame} & \text{else} \end{cases} \quad (9)$$

The parameters  $\delta_{low}$  and  $\delta_{high}$  can be tuned and will depend on the environment of the recording.  $\delta_{low}$  indicates a minimum required change in frame power, and  $\delta_{high}$  establishes an upper limit to discriminate against very loud sounds, such as a crashing chair or slammed door. Hence, direct human speech is considered to be limited within a range of power onset values.

**1.3.2 Speech-reverberant modulation ratio.** The speech-reverberant modulation ratio (SRMR) [9] is a metric that was developed towards predicting the intelligibility of speech in a given audio frame. Conceptually, anechoic speech (i.e., the direct signal) should have significant amplitude modulations between 4-16 Hz, which are related to the acoustic signals that correspond with syllables and phonemes. In the presence of reverberation, delayed and attenuated versions of this acoustic signal are summed together. This results in an increased level of envelope fluctuations at higher frequencies. Thus, a ratio of the modulations at low frequencies vs. those at higher frequencies provides a measure that is related to energy of the direct signal vs. that of the reverberant components.

We apply our own lightweight implementation of the SRMR, by first using the Hilbert transform to extract the envelope of the speech frame. The frequency content of the envelope is analyzed by computing the ratio of energy present in modulation bands associated with speech and modulation bands associated with reverberant audio content. The frequencies and bandwidths for the speech and reverberant bands are specified in [9]. Overall the frame classification is performed as follows for a frame  $F_i$ ,

$$F_i = \begin{cases} \text{speech frame} & \text{if } \delta_{high} > \frac{\sum_{j=1}^4 e_j}{\sum_{j=4}^8 e_j} > \delta_{low} \\ \text{non-speech frame} & \text{else} \end{cases} \quad (10)$$

where  $e_j$  is the energy present in the  $j$ -th frequency band of the extracted envelope. This ratio is used as a potential measure for voice activity, and is given thresholds  $\delta_{low}$  and  $\delta_{high}$  for similar reasons as the power onsets.

## 2 Problem statement

A number of methods have been introduced to perform the signal processing necessary for DOA estimation. These methods also involve numerical parameters, which will need to be selected for the human-robot interaction (HRI) task at hand. There is a need to identify the best parameters specifically for a binaural DOA setup on the REEM-C Humanoid Robot, which may be used in reverberant environments for the purposes of HRI. There is also a need to evaluate the implications of using these parameters in real-time, in terms of their accuracy and latency when it comes to HRI scenarios.

This work aims to tackle this problem by presenting a method to identify the best parameters for DOA estimation, including the classification methods, and numerical parameters such as frame sizes and thresholds. Parameters are optimized using a brute force and a Bayesian optimization approach, and used in a real-time implementation on the REEM-C, with a consideration for latency and potential applications for HRI.

## 3 Data preparation

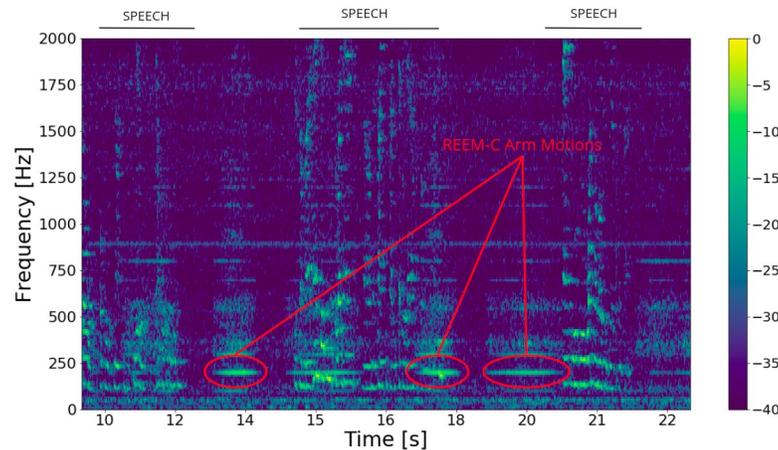
The binaural DOA setup is deployed onto the robot with a taut headband that places the microphones on the head of the REEM-C at the positions that would correspond with human ears, providing a realistic and human-like appearance and configuration. A Scarlett 2i2 audio interface was used with 2 lavalier microphones. This set up was chosen as it is inexpensive and could be adapted and deployed to a wide range of robotics platforms. The aperture of this array setup is measured from the calibration performed in section 1.2 as  $d = 0.255\text{m}$ .

Audio recordings were made in a lab environment with the robot operational. This simulates the noise that would be encountered while human-robot interaction scenarios are underway. The annotated periods of speech as well as ground truth locations of the speakers were used to properly estimate the parameters of the DOA pipeline in later sections. Audio recordings undergo no pre-processing steps; the annotations and signal processing techniques are made on the raw, untrimmed audio data. The specifics of each the recording conditions are shown in Table 1. Recordings were made in a variety of conditions. The speaker may be stationary or mobile, the sounds present may be speech or non-speech, and the robot may be either idle or active. An active robot performs numerous interactive gestures with the 7 degrees of freedom in the arms and the 2 degrees of freedom in the torso. An idle robot does not move, but is operational and therefore emits its own operational noise, which can be heard in the background of all of the recordings. Other non-speech sounds may include foot steps, typing of a keyboard and tapping of lab tools against table surfaces. This arrangement results in 8

**Table 1. Overview of the conditions recorded.**

SPEAKER BEHAVIOUR	SOUND SOURCE	ROBOT BEHAVIOUR	# RECORDINGS
stationary	speech	idle	5
stationary	speech	active	5
stationary	speech + non-speech	idle	5
stationary	speech + non-speech	active	5
mobile	speech	idle	5
mobile	speech	active	5
mobile	speech + non-speech	idle	5
mobile	speech + non-speech	active	5

<https://doi.org/10.1371/journal.pone.0296452.t001>



**Fig 6. Spectrogram of audio sample with human speech and REEM-C motions, represented in dB.** Three regions with energy that correspond to the sound from the REEM-C arm motors are highlighted.

<https://doi.org/10.1371/journal.pone.0296452.g006>

unique scenarios to perform recordings, and with 5 recordings made for each scenario, the final data set is of 40 audio recordings.

A good set of parameters will result in the classification that ignores the non-speech sounds but still accurately estimates the DOA of the human talker when they are speaking, even during the relatively noisy operations of the robot.

[Fig 6](#) shows the spectrogram of a sample recording, with simultaneous robot and speech sounds. The REEM-C performs some motions with the arms, which clearly show up in the spectrogram.

## 4 Optimization approaches

The optimization and parameter selection takes place via two methods, each with benefits and drawbacks. The results from both methods are compared and contrasted to assist with choosing the best set of parameters to use on the test set and the final robot implementation.

### 4.1 Brute force grid search

The brute force method is a simple approach to find ideal parameters for any mathematical function or optimization problem, and is a clear option to provide an initial set of baseline results. This baseline can then be evaluated against more comprehensive and innovative methods to better assess their overall contributions and value to the proposed problem. This method attempts every possible combination of parameters across the entire search space and chooses the parameters that best minimize the objective function. From [Table 2](#), the brute force method covers both classification methods (power onsets and the SRMR), 3 timing difference methods (time-domain cross correlation, GCC-PHAT and GCC-SCOT) and a series of numerical values (audio frame size, step size, low threshold and high threshold). To reduce computational time, trials are ended when no windows of speech are found, leading to empty DOA predictions. This is a computationally expensive approach since every possible combination of parameters will need to be attempted, and results will depend on the granularity of the defined search space.

**Table 2. Parameter spaces defined for both methods.**  $\mathcal{U}(\min, \max)$ = uniform distribution.  $\mathcal{N}(\text{mean}, \text{std})$  = normal distribution.

Parameter	Brute Force	TPE
Voice Method	SRMR, PO	SRMR, PO
Timing Method	cross-corr, gcc-phat, gcc-scot	cross-corr, gcc-phat, gcc-scot
Frame Size	(0.1, 1), step = 0.05	$\mathcal{U}(0.1, 1)$
Step Size (%)	(0.1, 1), step = 0.05	$\mathcal{U}(0.1, 1)$
Low Threshold	(1,10), step = 0.1	$\mathcal{N}(3,3)$
High Threshold	(3,14), step = 0.1	$\mathcal{N}(10,3)$

<https://doi.org/10.1371/journal.pone.0296452.t002>

## 4.2 Tree Structured Parzen Estimator

A second optimization method chosen for this work is the Tree Structured Parzen Estimator (TPE), which is a Bayesian optimization approach that evaluates past results to generate a probabilistic model of the hyperparameters and associated score. This probabilistic approach is better suited for the proposed task as opposed to methods such as a random search, which can provide highly variable results. Random search, while less computationally intensive than the grid search, suffers from the risk of wholly ignoring the ideal set of hyperparameters because of its uninformed, random approach. The chosen Bayesian method takes an informed approach, and can have a higher chance of producing better parameters while not requiring a search of the entire parameter space. In addition, this method can search values on continuous distributions, and is not limited by a granularity, such as the step size for each parameter, that may be defined by the user. This method is therefore more suitable to compare to the baseline results produced by the grid search approach described above.

Given a series of objective values, *score*, with their respective parameters, *parameters*, the TPE method generates two probability distributions by segmenting the results based on a threshold *score\**.

$$p(\text{parameters}|\text{score}) = \begin{cases} l(\text{parameters}) & \text{if } \text{score} < \text{score}^* \\ g(\text{parameters}) & \text{if } \text{score} \geq \text{score}^* \end{cases} \quad (11)$$

The method then selects parameters with a greater probability of being under  $l(\text{parameters})$  than  $g(\text{parameters})$ , given that  $l(\text{parameters})$  is built from trials with more favourable objective values. This informed reasoning is used to select the next set of hyperparameters while updating the two distributions, allowing the method to find an optimal set of parameters while not exhaustively searching the entire parameter space. The TPE is implemented via the hyperopt package [10].

Given the parameter space in Table 2, the key difference from the brute force method is that the numerical parameters (frame size, step size, low and high thresholds) are now placed on a continuous distribution. The uniform distribution for the frame size and step size ensure each value has an even chance of being selected. The normal distribution parameters for the low and high thresholds are chosen based on a few tests that may indicate where good thresholds may lie, given that the high threshold must be larger than the low threshold. Table 2 outlines the parameter spaces searched for both methods.

### 4.3 Objective functions

In order to properly define this optimization task, key variables are first defined.

$$f1(\omega) = 2 * \frac{precision(\omega) \cdot recall(\omega)}{precision(\omega) + recall(\omega)} \quad (12)$$

$$mse(\omega) = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i(\omega))^2$$

The vector  $\omega$  represents all input parameters for any single trial, which come from Table 2. The F1-score and mean squared error are then computed for a single trial with a set of parameters  $\omega$  as per Eq 12. The F1-score requires the computation of the precision and recall, which are common metrics used to evaluate the performance of a classification algorithm by computing the true positives, false positives, true negatives and false negatives. The F1-score is then the harmonic mean of the precision and recall, and is often used in machine learning tasks as a performance metric.

We split the data set into a training and test set, with a split of 75% to 25% respectively. Hence, out of 40 recordings, 30 are randomly chosen for the training portion, and the remaining 10 are used for the testing portion. To ensure proper scientific protocol, the objective functions are only used for the training examples, whereas the test examples will be evaluated on using the results of the optimizations. A random selection for the training set ensures each of the 8 unique scenarios can be well represented in the optimization pipelines. The random selection results in trials 3, 5, 6, 7, 10, 18, 28, 29, 30 and 36 being reserved for the test set, whereas the remainder are used for training.

The objective function used in the optimization approaches varies for each problem. The performance of the DOA estimate classification must be good, and as a result of potential imbalances in the dataset, the F1-score for classification is considered the metric to optimize. Hence the objective function is formulated for classification as follows, which computes the F1-score for every  $j$ -th trial, and aims to minimize the negative of the weighted average across all  $M$  recordings used for the optimization.

$$\begin{aligned} \min_{\omega} \quad & - \frac{\sum_{j=1}^M f1(\omega)}{M} \\ \text{s.t.} \quad & 0 < \delta_{low} < \delta_{high} \end{aligned} \quad (13)$$

For DOA estimation, the mean squared error is considered as the objective to minimize as the generated estimates and ground truth are continuous variables. The objective for DOA estimation computes the average mean squared error across all  $M$  trials used for optimization.

$$\begin{aligned} \min_{\omega} \quad & \frac{\sum_{j=1}^M mse(\omega)}{M} \\ \text{s.t.} \quad & 0 < \delta_{low} < \delta_{high} \end{aligned} \quad (14)$$

We also explore how to perform both optimizations at once in a joint manner. The joint optimization aims to minimize the MSE for DOA estimation and maximize F1-score for classification. The objective for this method is formulated accordingly, using the two metrics as a

fraction. Eq (15) shows this formulation.

$$\begin{aligned} \min_{\omega} \quad & \frac{\sum_{j=1}^M \frac{mse(\omega)}{f1(\omega)}}{M} \\ \text{s.t.} \quad & 0 < \delta_{low} < \delta_{high} \end{aligned} \quad (15)$$

A modification is added to regularize the frame size  $\gamma$  during the optimization. Theoretically, this should result in lower frame sizes found with good results on both DOA and classification tasks, meaning potentially lower latencies when used on the robot for real-time operation. The value of  $\lambda$  is set to 0.5 for this work. This value reflects the importance placed on the frame size parameter when computing the objective value. With a value as high as 0.5, we effectively state that the frame size is half as important as the performance metrics when computing the objective value, and hence may lead to more favorable parameters with lower frame sizes. In addition, since the frame size parameter varies from 0-1, a magnitude of 0.5 appears reasonable for  $\lambda$ , such that the frame size is not too heavily or too lightly weighted. This value also has a similar magnitude as metrics included in the objective function, such as the F1-score which also ranges from 0-1. We believe that slightly varying this regularization parameter would likely still lead to similar results as presented in the work thus far. This objective function will be helpful to investigate the effect of frame sizes on the final results. Eq (16) shows this regularized formulation.

$$\begin{aligned} \min_{\omega} \quad & \frac{\sum_{j=1}^M \frac{mse(\omega)}{f1(\omega)}}{M} + \lambda|\gamma| \\ \text{s.t.} \quad & 0 < \delta_{low} < \delta_{high} \end{aligned} \quad (16)$$

## 5 Results

We present results for classification, DOA and the joint performance with both parameter search methods. Qualitative evaluation was also conducted on the chosen parameters, and other considerations not included in this optimization are discussed.

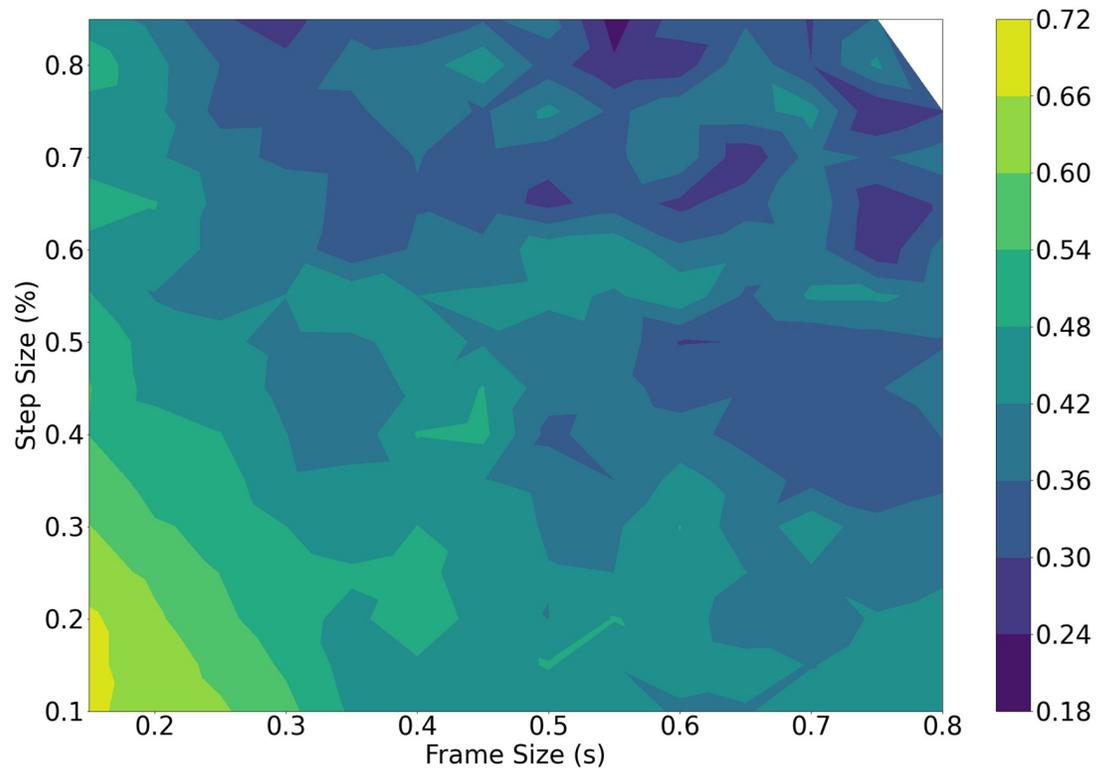
Initial quantitative results are presented via contours for visualization. Since there are a total of 6 dimensions to this problem, not all trends can be visualized. These results are further explored below in table form as well.

### 5.1 Brute force method

**5.1.1 DOA accuracy.** The brute force method results are shown in Fig 7 comparing the frame size and step size to the weighted average MSE as a contour plot. The minima, shown as dark regions, occur primarily with larger frame sizes and larger step sizes. The performance on the DOA tends to worsen as the frame size or step size are reduced, indicating that the best choice for this task may require larger audio chunks when used in real-time.

A tendency towards a higher step size also indicates that it is less important to capture overlapping audio information, as the subsequent ITD estimation is still able to generate good results.

**5.1.2 Classification accuracy.** Classification results with power onsets do not exceed an F1-score of 20%, whereas the SRMR performs far better, giving maximum results 70%. The relationship between the thresholds and the classification performance is simple to interpret, as the best results are consistently obtained with a low threshold around 1.5. The high



**Fig 7. Brute force DOA performance against frame size (s) and step size (%).** The small white region corresponds to parameter combinations that resulted in no windows of speech being detected.

<https://doi.org/10.1371/journal.pone.0296452.g007>

threshold appears to be less important, and can be set to 7 to get good classification results. Fig 8 demonstrates the relationship of the classification performance to the set thresholds. The best results are generated using the SRMR as the classification method, with a clear maxima around the specified low threshold of around 1.5 and a high of around 7. Parameters that generated no results, as they detected no windows of speech, show up as white regions in the contours.

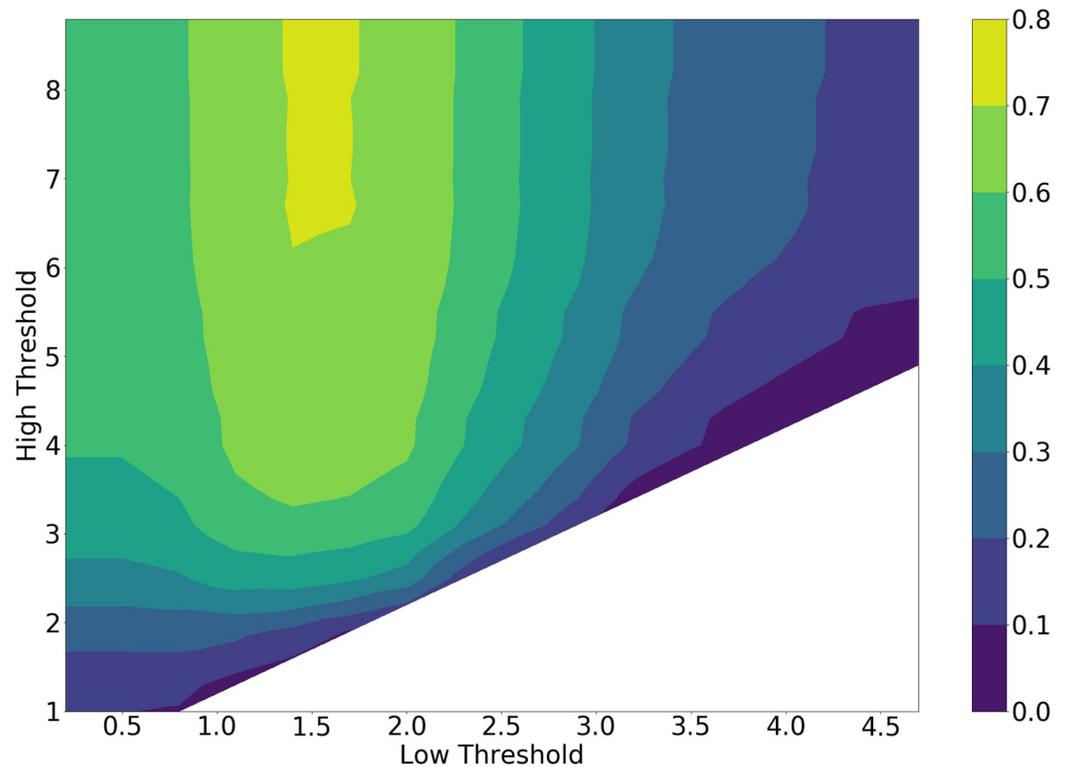
## 5.2 TPE performance

The TPE method is evaluated on all objective tasks next. The TPE method is run for 1000 iterations and completes within a few minutes for each case, highlighting the computational efficiency of this technique while still searching the parameter space in an informed manner.

**5.2.1 DOA accuracy.** The results on the DOA task are shown in Fig 9 as a contour plot.

Naturally, TPE performs far fewer iterations and thus outputs fewer data points to visualize. The best results are consistently found using the GCC-PHAT as a timing difference method, and a mixture of the SRMR and power onsets as classification methods. The contour plot shows lower MSE values for larger frame sizes and step sizes, similar to the brute force results. Certain frame sizes and step sizes are never sampled by the estimator since they do not indicate a high probability of generating a good score, leaving white areas on the contour.

**5.2.2 Classification accuracy.** We run the Bayesian optimizer for 1000 trials to optimize the task of detecting the presence of speech. The results are shown against the frame size and step size in Fig 10.



**Fig 8. Brute force classification performance against low and high thresholds.** The white region corresponds to parameter combinations that resulted in no windows of speech being detected.

<https://doi.org/10.1371/journal.pone.0296452.g008>

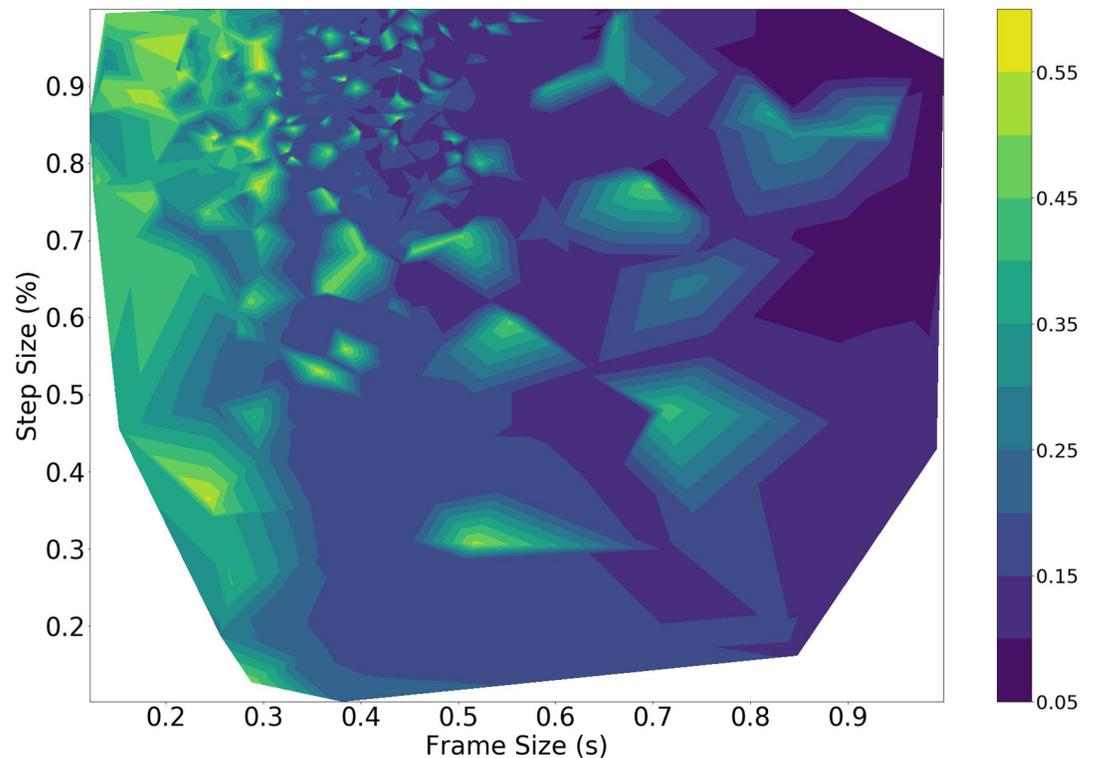
A number of maxima in the contour plot are seen with an F1-score around 70%, which are generated using the SRMR as the main method for classification. The optimizer is unable to find good classification results for the power onsets, as the maximum F1-score is around 20%. This is unsurprising, as this metric will only select a good frame if its power exceeds the previous frame's power by a certain factor. For periods of continued speech, the subsequent frame-to-frame power ratio will not be very high, and so a large amount of audio frames containing speech will be rejected.

The relationship of the frame size and step size to the classification performance is more difficult to establish, as compared to the thresholds in Fig 10.

**5.2.3 Joint optimization results.** The individual tasks generate different results for the best set of parameters  $\omega_{best}$ . In order to accomplish both tasks effectively, the joint objective function will need to be optimized. The joint objective function is run through the same TPE optimization pipeline for 5000 iterations and generates results as in Fig 11.

The best loss values are generated for frame sizes closer to 1s while optimizing for both the DOA and classification performance. Results are consistently best using the SRMR and GCC-PHAT. The GCC-SCOT appears sparsely in the full results, indicating that the optimizer does not find this technique to be as effective as the GCC-PHAT, and so does not tend to apply it during the learning process. These results tend to agree with what is seen in Figs 10 and 11, as the minima occur with large frame sizes. It seems to be clear that as the frame size increases, the joint optimization results improve as per the trends seen in the presented contour.

In the context of real-time performance on a robot, larger frame sizes will require more time to generate a response for reorientation by the robot. In order to provide a realistic



**Fig 9. TPE DOA performance against frame size (s) and step size (%).** The white region corresponds to parameter combinations that were not tested by the TPE method.

<https://doi.org/10.1371/journal.pone.0296452.g009>

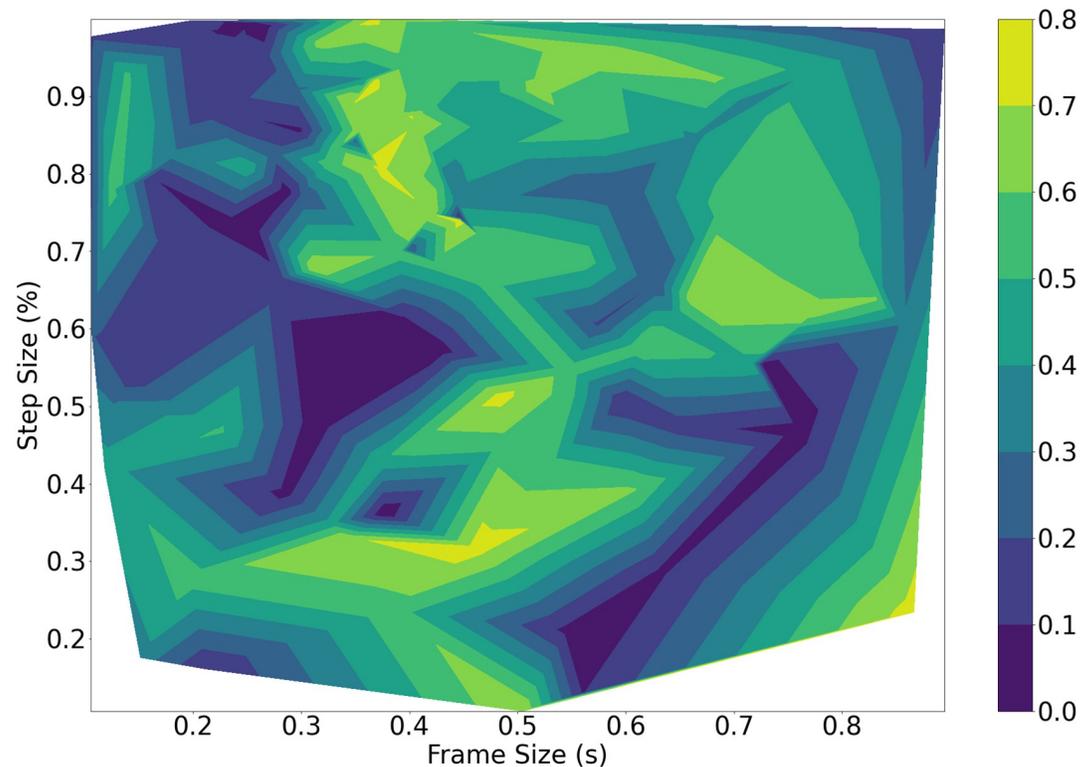
human-robot interaction, the system should be able to detect and respond within 200-300 ms. Therefore, it is desirable to achieve good classification and DOA performance with lower frame sizes. The optimization is performed again via the TPE method with the regularized objective function, and generates the results as in Fig 12.

As per the contours, more minima are concentrated around the 300-450 ms range for frame sizes. The previously chosen frame sizes above 400 ms are now no longer producing minimal objective values, as they would be penalized by the regularization parameter.

Further results in Fig 13 show how the average objective values change with regards to the frame size, for the regularized joint objective. With no regularization in the learning process, the larger frame sizes at 700 ms or above tend to have lower objective values, which corresponds to previous results. With regularization added, lower frame sizes are favoured, leading to a lowest average objective value at a size of 350 ms, as shown in Fig 14.

The results for all the optimization tasks with either method are shown in Table 3. The frame size and step size are reported as they are more crucial to the operation of the robot in real-time. The previous classification results indicate that a good selection for the low and high thresholds is 1.5 and 7, respectively. Overall, the choice of these thresholds is less consequential as their value will not affect the latency of the robot when used in real-time.

Table 3 also shows the MSE and F1-score results when different metrics are minimized. For instance, when looking for the best joint objective value, the TPE method yields an MSE of 0.08 and an F1-score of 0.70. In contrast, when looking for the best classification performance, the F1-score is 0.84, with a much higher MSE of 0.87.

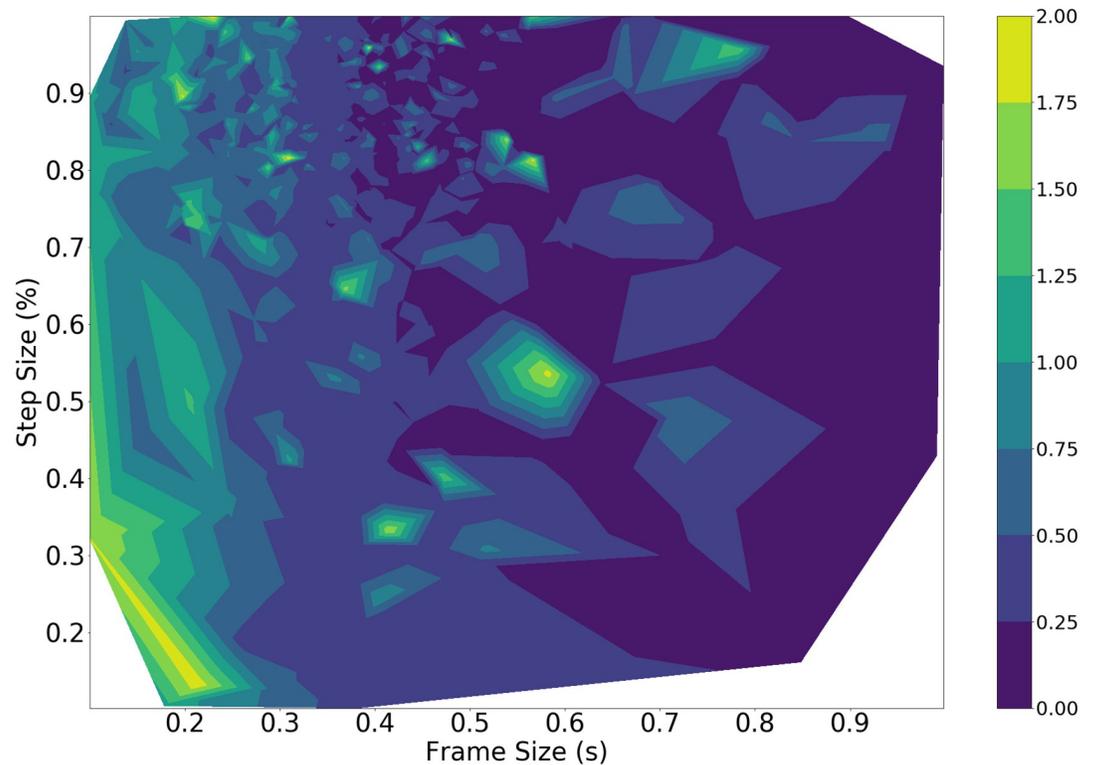


**Fig 10.** TPE classification performance visualized against frame size (s) and step size (%). The white region corresponds to parameter combinations that were not tested by the TPE method.

<https://doi.org/10.1371/journal.pone.0296452.g010>

It is important to note that the brute force method is limited in its search as it can only evaluate discrete numerical parameters, whereas the TPE method can choose values from continuous distributions. This will have an effect on how the TPE method learns.

Given the presented results, and the quantitative results shown in Table 3, the best parameters for  $\omega_{best}$  using both optimization methods are found in Table 5. These parameters reflect those that produced the most favourable objective values; the joint regularized task with the TPE optimizer found a good MSE of 0.11 and a relatively high F1-score of 0.69, while also producing a more favourable frame size than the remaining results, and hence were chosen to be the best across the entire study. The joint optimization without regularization with the TPE method produces a similar F1-score of 0.70 with a lower MSE of 0.08, but this comes at the cost of a very large frame size at 919 ms, which is unfavorable for the real-time use case on the robot. Similarly, a trade-off is seen for the performance across the results between the F1-score, MSE and frame size, and hence the best parameters are taken from the TPE method with the joint regularized optimization results. We consider an F1-score of 0.69 acceptable, considering the numerous non-speech and active robot sounds present in many of the training set recordings that may interfere with the voice classification problem. The best low and high thresholds for the brute force method are found from the voice classification results shown in Fig 8 and are taken to be 1.5 and 7 respectively. The best thresholds vary greatly for the TPE method given its continuous sampling approach, but the low and high values that produce the best objective values are found to be 2.2 and 11 respectively. For the sake of interpretation and evaluation, the best frame size of 318 ms was adjusted to 320 ms.



**Fig 11. Joint objective loss vs. step size and frame size.** The white region corresponds to parameter combinations that were not tested by the TPE method.

<https://doi.org/10.1371/journal.pone.0296452.g011>

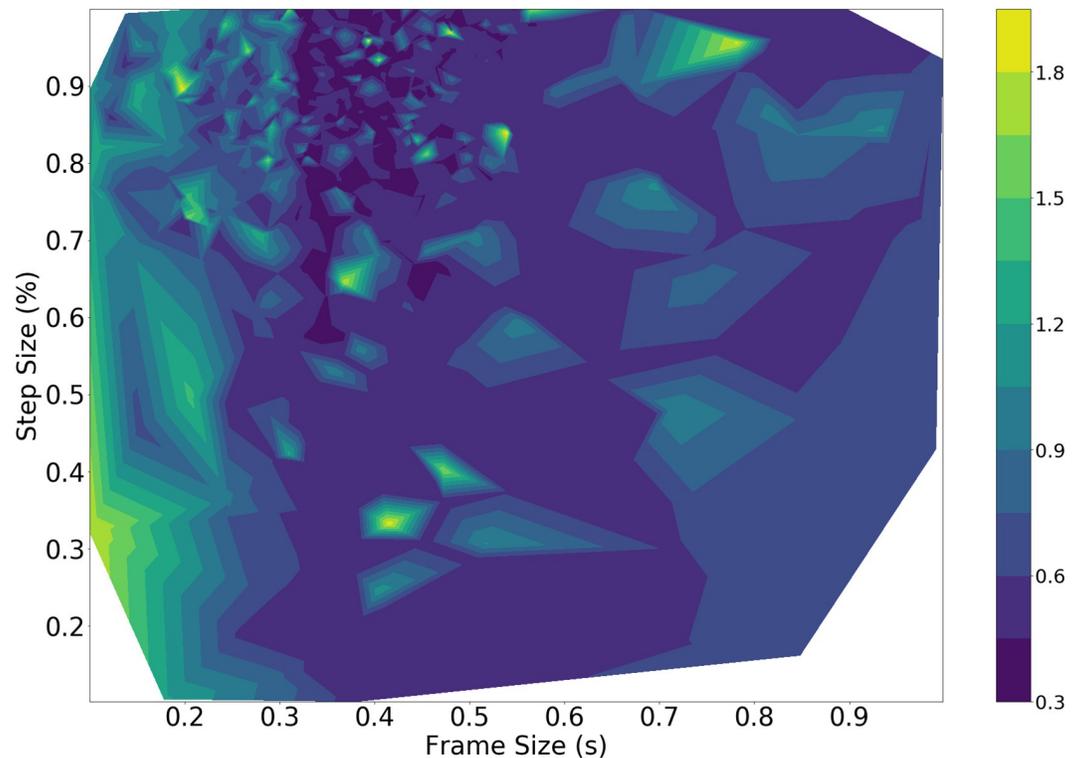
Additionally, these best parameters from each method are applied to the test set and generate results as in Table 4. These show the best MSE and F1-score results using the best results from both optimization methods.

As per Table 4, the performance of both sets of parameters on the test recordings are quite similar to the results on the training set. This indicates that the parameters have not been overfit to the training set, are more generalizable and can be robust to a variety of acoustic scenarios. Despite the slight benefit to the F1-score from the brute force parameters, the TPE results have a significantly better MSE for DOA estimation and uses a frame size that is 80 ms smaller than the brute force result. Hence, the best parameters from the TPE optimization are chosen to be the best result across the entire study.

An example of the test set results are shown in Figs 15–17 for 3 sample test recordings using the best TPE parameters found in Table 5. The details about the acoustic conditions for each test recording are presented in the data file attached with this submission.

We also investigate how the voice classification and timing difference methods perform in relation to each other. We study the best trials found in the optimization results for every combination of the voice method and timing method, to further highlight how the use of these different methods affects the pipeline's performance. Table 5 reports on the MSE and F1 score for each of these combinations separately, for the top 50 and the top 100 trials in the optimization results.

With respect to the voice classification methods, the SRMR is clearly the better performer with regards to its MSE and F1-score as compared to the power onsets. It produces significantly better results for both the top 50 and the top 100 trials, whereas the power onsets overall

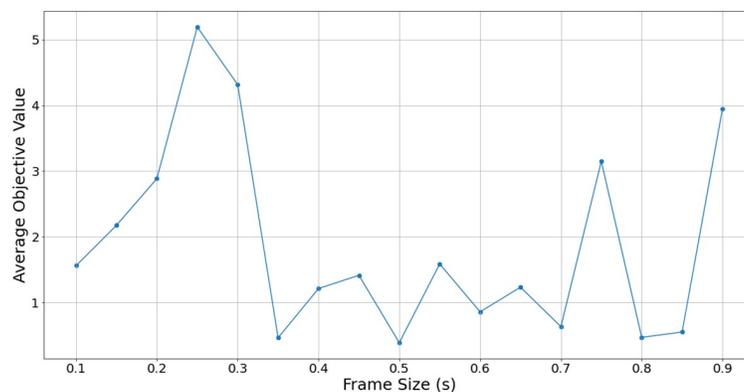


**Fig 12. Joint regularized objective loss vs. step size and frame size.** The white region corresponds to parameter combinations that were not tested by the TPE method.

<https://doi.org/10.1371/journal.pone.0296452.g012>

seem to be unreliable for this problem. This further explains why both the brute force and the TPE methods unanimously choose the SRMR as the best option.

The timing method results indicate that the standard beamformer is incapable of handling this problem well, as it suffers from poor performance across the entire study, primarily due to high error in DOA estimation. The GCC-PHAT and GCC-SCOT appear to be much better at performing the DOA estimation, and have similar MSE and F1-scores in the top 50 and top 100 results. This would suggest that they both could be suitable to solving this problem.



**Fig 13. Frame size vs. average joint objective loss.**

<https://doi.org/10.1371/journal.pone.0296452.g013>

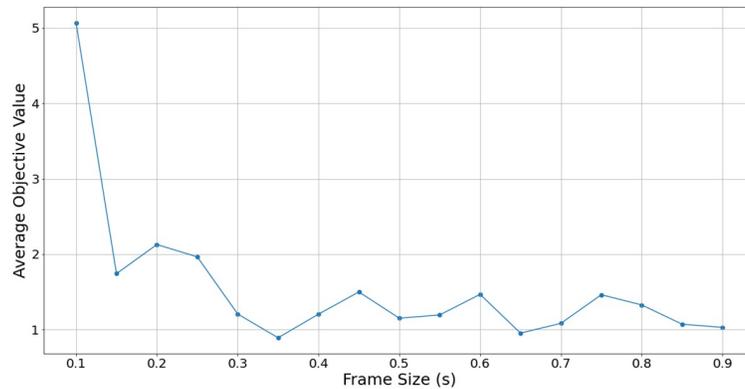


Fig 14. Frame size vs. average joint regularized objective loss.

<https://doi.org/10.1371/journal.pone.0296452.g014>

Table 3. Best parameters and results for each task across optimization methods.

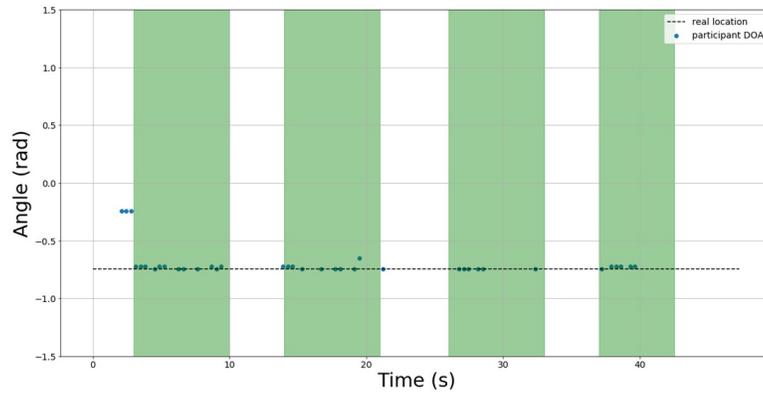
Task	DOA	Classification	JOINT	JOINT + REG
<b>Brute Force</b>				
Voice Method	PO	SRMR	SRMR	SRMR
Timing Method	GCC-PHAT	GCC-PHAT	GCC-PHAT	SCOT
Frame Size	850	400	500	400
Step Size	0.90	0.50	0.55	0.90
MSE	0.05	0.66	0.19	<b>0.16</b>
F1-score	0.14	0.69	0.67	<b>0.67</b>
<b>TPE Optimizer</b>				
Voice Method	SRMR	SRMR	SRMR	SRMR
Timing Method	GCC-PHAT	SCOT	GCC-PHAT	GCC-PHAT
Frame Size	896	627	919	318
Step Size	0.97	0.25	0.92	0.93
MSE	0.06	0.87	<b>0.08</b>	<b>0.11</b>
F1-score	0.70	0.84	<b>0.70</b>	<b>0.69</b>

<https://doi.org/10.1371/journal.pone.0296452.t003>

Table 4. Test set performance.

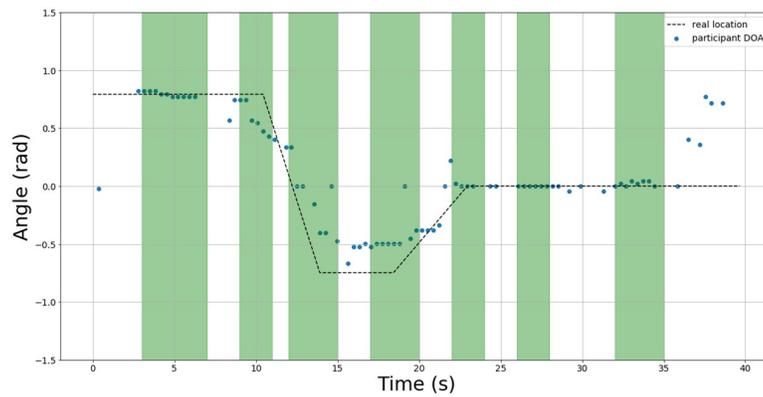
Test Recording	TPE—MSE	TPE—F1-Score	Brute Force—MSE	Brute Force—F1-Score
Test 3	0.001	0.82	0.001	0.84
Test 5	0.651	0.65	0.819	0.77
Test 6	0.082	0.56	0.097	0.67
Test 7	0.026	0.72	0.022	0.80
Test 10	0.196	0.68	0.236	0.77
Test 18	0.019	0.72	0.025	0.74
Test 28	0.056	0.67	0.052	0.67
Test 29	0.054	0.72	0.105	0.74
Test 30	0.098	0.77	0.159	0.77
Test 36	0.036	0.69	0.093	0.65
<b>Average</b>	<b>0.102</b>	<b>0.70</b>	<b>0.161</b>	<b>0.74</b>

<https://doi.org/10.1371/journal.pone.0296452.t004>



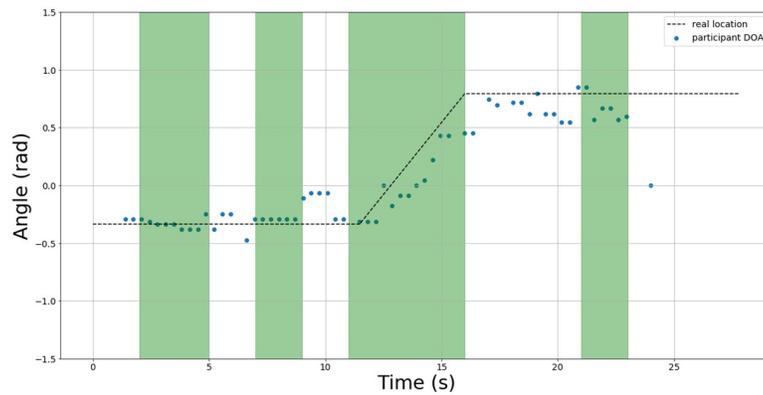
**Fig 15. Test 6 DOA results.** The green regions indicate intervals where speech was present. The blue dots indicate the estimated DOA using best study parameters. The actual DOA of the speaker is indicated by the dotted line.

<https://doi.org/10.1371/journal.pone.0296452.g015>



**Fig 16. Test 28 DOA results.** The green regions indicate intervals where speech was present. The blue dots indicate the estimated DOA using best study parameters. The actual DOA of the speaker is indicated by the dotted line.

<https://doi.org/10.1371/journal.pone.0296452.g016>



**Fig 17. Test 36 DOA results.** The green regions indicate intervals where speech was present. The blue dots indicate the estimated DOA using best study parameters. The actual DOA of the speaker is indicated by the dotted line.

<https://doi.org/10.1371/journal.pone.0296452.g017>

Table 5. Comparative results for voice and timing methods.

Voice Method	Timing Method	Top 50 MSE	Top 50 F1-Score	Top 100 MSE	Top 100 F1-score
SRMR	Beamformer	0.67	0.72	0.93	0.61
SRMR	GCC-PHAT	0.11	0.71	0.11	0.70
SRMR	GCC-SCOT	0.11	0.70	0.12	0.70
PO	Beamformer	0.91	0.32	1.04	0.38
PO	GCC-PHAT	0.41	0.66	0.84	0.54
PO	GCC-SCOT	0.66	0.45	0.89	0.37

<https://doi.org/10.1371/journal.pone.0296452.t005>

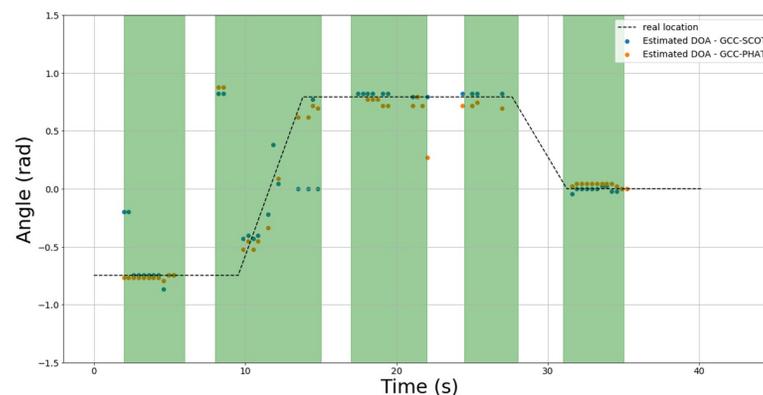
However, when evaluating the 50 best trials from the optimization overall, the GCC-SCOT only appears twice, and the GCC-PHAT is present in the other 48 trials. The GCC-PHAT also provides a better MSE when paired with the power onsets as shown in Table 5, indicating it is more robust than GCC-SCOT. This would indicate that the GCC-PHAT is the better choice for this pipeline, and corroborates the results found in Table 3, where it seems to be the timing method of choice for nearly all of the presented conditions.

We look at how the two timing methods differ in greater detail, as the results they generate are quite similar in the optimization results. Fig 18 shows an example of results on test 29 using two sets of estimates; one generated with the GCC-PHAT and the other generated with the GCC-SCOT.

Overall, the estimates from the two methods are quite similar for the trial. However, it appears the GCC-SCOT estimates stray from the ground truth more often, as seen by some of the dots near the beginning of the trial and around 13 s. Since trial 29 involves the active robot condition, this may be an indication that the GCC-SCOT is less robust to the sounds produced by the robot's gestures. This is further evidence that the GCC-PHAT is the overall best choice for the timing delay estimation process.

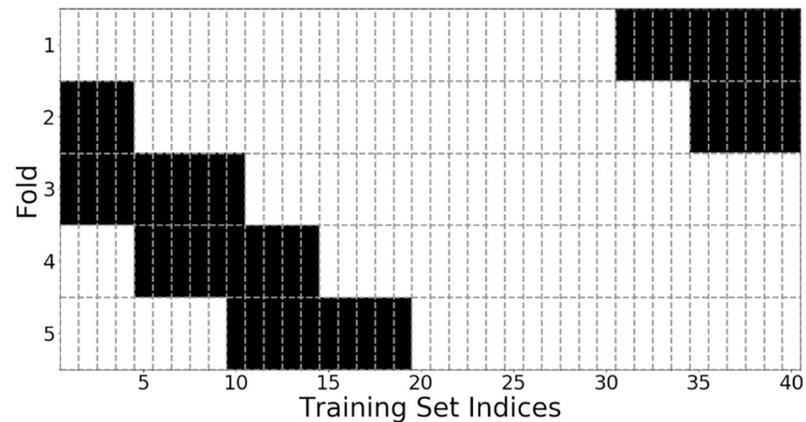
## 6 Cross validation approach

We take a cross validation approach for the optimization in order to further evaluate the results and ensure they are reliable for different recordings used as training data. This can help avoid a selection bias and ensure that the optimized parameters properly generalize to the



**Fig 18. Test 29 DOA results.** Estimates are generated comparing the GCC-PHAT (orange) and the GCC-SCOT (blue). The green regions indicate intervals where speech is present. The actual DOA of the speaker is indicated by the dotted line.

<https://doi.org/10.1371/journal.pone.0296452.g018>



**Fig 19. Visualization of 5 dataset folds used for cross validation.** Indices shaded white are used for training, and indices shaded black are used for testing.

<https://doi.org/10.1371/journal.pone.0296452.g019>

variety of acoustic conditions recorded in the dataset. We do this by generating 5 folds of the dataset, each fold containing 30 recordings, and running the TPE optimization pipeline on each fold. It would be inefficient to use the brute force method for this validation due to its exhaustive nature, which further highlights the ability of the TPE to perform informed optimization in a time-efficient manner. Fig 19 shows the training set indices used for all 5 folds of the data. To ensure 30 recordings are present in each fold, the indices wrap around once they reach the end of the dataset, as shown for folds 4 and 5.

Each fold is optimized using the regularized joint objective for 1000 trials and the results are collected and presented in Table 6 below. The table also shows the performance of the parameters on the test set of each specific fold.

Evidently, each fold has produced similar results for nearly all parameters. The SRMR and the GCC-PHAT are overall most preferred for the voice classification and timing difference estimation methods, and frame sizes tend to be around 300-400 ms. This is in agreement with the location of the minima shown in Fig 12. The 5th data fold prefers the power onset as the voice classification method and has a smaller step size of 0.27. We believe this is due to fold 5 containing the most recordings with a mobile sound source, which may require the pipeline to use a greater overlap to evaluate successive audio frames.

Overall, this cross validation confirms that the parameters found on the original optimization are robust to a variety of conditions and can be used as a general set of parameters to estimate DOA on the REEM-C. Each fold has a variety of unique acoustic conditions represented in its data, but due to the relative similarity of the optimization's results on all the folds, the presented experimental results are further validated. Due to its even representation of the

**Table 6. Cross validation results.**

Fold	Voice Method	Timing Method	Frame Size	Step Size	Test MSE	Test F1-Score
1	SRMR	GCC-PHAT	342	0.94	0.12	0.72
2	SRMR	GCC-PHAT	378	0.95	0.13	0.69
3	SRMR	GCC-SCOT	422	0.78	0.22	0.67
4	SRMR	GCC-PHAT	384	0.77	0.14	0.69
5	PO	GCC-PHAT	330	0.27	0.16	0.69

<https://doi.org/10.1371/journal.pone.0296452.t006>

Table 7. Best overall parameters for both methods.

Optimizer	Classification Method	ITD Method	Frame Size	Step Size	Low Threshold	High Threshold
TPE	SRMR	GCC-PHAT	320	0.93	2.2	11
Brute Force	SRMR	GCC-SCOT	400	0.90	1.5	7

<https://doi.org/10.1371/journal.pone.0296452.t007>

acoustic conditions, we choose to use the TPE method's results found from the optimization on the randomized data as shown in Table 7, and select these to be the best parameters  $\omega_{best}$  for this whole study.

## 7 Use on real robot

We deploy this system onto the REEM-C Humanoid for use in real-time using a full ROS integration. Audio frames are saved to a buffer and processed with the parameters generated from  $\omega_{best}$ , allowing for real-time estimation of DOA. Fig 20 shows the REEM-C's head with microphones installed.

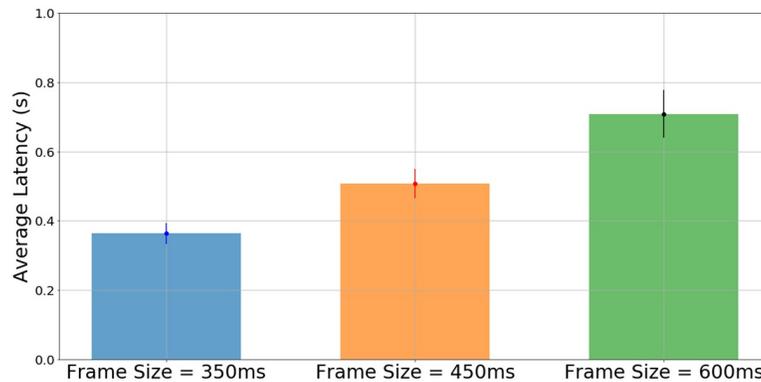
We also study the effects of latency on the performance of the real-time tracking and estimation. Experiments are carried out with frame sizes of 350ms, 450ms and 600ms, with all other parameters kept constant. The measured DOA are recorded, as well as the audio power level measured by a separate USB microphone, along with the timestamps for both metrics as measured by the ROS network. This allows for identifying how long it takes for a DOA to be measured once the onset of speech has been detected. Fig 21 shows the average latency measured with the 3 frame sizes.

The latency is measured to have an average of 0.363s with a standard deviation of 0.076s for the 350ms frame size, an average of 0.508s with a standard deviation of 0.103s for the 450ms frame size, and an average of 0.708s and a standard deviation of 0.169s for the 600ms frame size. For 10 separate DOA measurements at 350ms and 600ms, a two-tailed t-test for their



Fig 20. Microphone setup on REEM-C.

<https://doi.org/10.1371/journal.pone.0296452.g020>



**Fig 21. Average latencies of DOA estimate for three different frame lengths.** The errorbars indicate one standard error.

<https://doi.org/10.1371/journal.pone.0296452.g021>

latencies yields a p-value of 0.0019. With  $\alpha$  set to 0.05, this indicates that the choice of frame size is indeed significant for real-time use, further validating the regularization applied in the optimization and the choice of lower frame sizes for  $\omega_{best}$ .

## 8 Discussion

The generated results across all methods have noticeable similarities and differences. For the DOA task, as per Figs 7 and 9, the brute force and the Bayesian methods mostly lead to frame sizes that are 500 ms or larger, and step sizes larger than 75%. The GCC-PHAT succeeds most often as a method to estimate timing difference compared to the standard beamformer and the GCC-SCOT. The best frame selection method happens to be the power onset for the brute force method. This is unsurprising as power onsets should be dominated by energy from the direct path (i.e., have a high direct to reverberant energy ratio) and so will likely produce accurate DOA estimates when the power onset condition is met. However, this comes with the trade off of rejecting many other frames containing speech as the subsequent frame-to-frame power ratio during periods of continuous speech can be similar. This potentially ignores many frames where the ratio of direct to reverberant energy may still be high, and this explains the poor voice classification performance of the power onsets.

This is more evident when optimizing for the classification task. Both optimization methods point towards using the SRMR as the main method to detect periods of speech, with classification based on power onsets consistently producing poor results (no more than 20% F1-score).

Since both tasks produce different results, the joint objective results should be investigated to identify parameters that perform well for both the classification and DOA tasks. The joint objective task for both methods favours the SRMR and GCC-PHAT for processing the audio frames. In addition, the brute force method results in a frame size of 500 ms, whereas the TPE method finds the best results to occur with a frame size of 919 ms. These results both require a long latency when implemented on a robot—orienting behaviour on the robot will lag any movement of a talker by at least half a second.

Studies in turn-taking dynamics and conversational behaviour indicate that humans take on average 200-300 ms to respond to their partners [11], suggesting that lower frame sizes will be more required for more natural for HRI behaviour. The joint regularized task produces the best results with lower frame sizes, as is depicted by where the minima lie on the contour plot

in Figs 11 and 12. The best results for the study are then taken from the joint regularized task using the TPE method given its performance on the test set. It is important to note that regardless of how the optimization is performed, good results are rarely found for both tasks with frame sizes less than 300 ms, as shown by where the minima lie in Fig 12. We suspect that this is due to the calculations involved in computing the SRMR. For the SRMR, the process involves studying the modulation of the speech signal via its envelope, and extracting the energies present in certain bands of this envelope. The lowest frequency band for this metric was centred at 4Hz. Thus, one period of this modulation corresponds with 250 ms. Frame sizes shorter than this length may result in inaccurate estimation of the 4 Hz component of the modulation energy. Thus, the use of SRMR as it is was defined here may impose a minimum latency that is too long to achieve human-like behaviour. While increasing the minimum modulation frequency used in the SRMR would reduce the minimum latency, further work is needed to determine the effect this would have on classification performance. If other classification methods are explored, minimum latencies should be less than 200 ms.

Table 3 gives further insight to the numerical performance of these methods. The values indicate differences in performance when searching for different objective value minima. Numbers in bold indicate the notable differences in performance when optimizing for either objective value. For instance, using the brute force method, searching for the minimal regularized objective provides a slightly better MSE of 0.16 compared to the unregularized objective with the same F1-score of 0.67. In contrast, the TPE method sees a slight increase in MSE from 0.08 to 0.11, and a slight decrease in F1-score from 0.70 to 0.69 when minimizing the regularized objective as opposed to the unregularized objective. However, this minimal decrease in performance comes at a frame size that is much smaller (318 ms vs. 919 ms). This is supporting evidence that the joint objective function was appropriate for this problem as both classification and DOA tasks are performed to an acceptable level with the best parameters  $\omega_{best}$ .

The results on the test set show that these parameters are reasonable and have not overfit on the training set, as the average performance on both tasks are good for all test set recordings. We also see that the TPE method generates significantly better MSE of 0.102 compared to the brute force method MSE of 0.161, but slightly worse F1-score, as per Table 4. We suspect that the higher step size and different thresholds from the brute force method provide a greater resolution for identifying speech on the microphone signals, leading to slightly better classification performance on test data.

Overall, the test recordings for both moving and stationary speakers produce good results. The pipeline also seems to be robust to active robot conditions and non-speech sounds, evidenced by good performance on tests 10, 18, 28, 29, 30 and 36. It appears test 6 has worse performance on the F1-score than the other recordings; we suspect this is because the active robot introduced a large amount of background noise that made this trial especially difficult. However, it appears that the background noise did not interfere with the DOA estimation, as the MSE for this trial was still found to be low, hinting at the abilities of the GCC-PHAT to effectively filter out unwanted reverberation. We believe the MSE for tests 5 and 10 is higher than the rest because of the angle at which the speaker was present relative to the robot. Certain motions of the robot require its arms to move along the side of the head, which likely has added interference to the direct path of the speaker on these two recordings.

With a functioning sound source localization pipeline in real-time, the potential for HRI can be expanded. For instance, if moving conversational partners can be detected by the robot, HRI can be augmented by implementing a human-like, realistic tracking behaviour. Motion capture analysis and modeling of the head, shoulders and feet such as in [12] can be applied for this purpose.

## 9 Conclusions

This work presented a pipeline to perform binaural direction of arrival estimation on a humanoid robot. A dataset of 40 unique recordings was collected, encompassing 5 recordings each of 8 unique conditions involving the speaker, the robot, and the nature of the sounds produced. Optimization procedures were used to improve the performance on several trials in the acoustic environment of the robot, and can find consistent results regarding the best classification and ITD methods, including the relevant numerical parameters. The TPE method produced the best overall results from the training data, and thus its optimized parameters were chosen as the optimal ones for the entire study. Test set results indicate that the chosen parameters are appropriate for a variety of acoustic scenarios specific to human-robot interaction, such as a moving human conversational partner, or a robot that is performing gestures during the real-time interaction. A method to use this pipeline on the real REEM-C is also presented, with considerations for real-time latency and performance.

## Author Contributions

**Conceptualization:** Pranav Barot, Katja Mombaur, Ewen N. MacDonald.

**Formal analysis:** Pranav Barot, Ewen N. MacDonald.

**Investigation:** Pranav Barot.

**Methodology:** Pranav Barot, Ewen N. MacDonald.

**Software:** Pranav Barot.

**Supervision:** Katja Mombaur, Ewen N. MacDonald.

**Visualization:** Pranav Barot.

**Writing – original draft:** Pranav Barot.

**Writing – review & editing:** Katja Mombaur, Ewen N. MacDonald.

## References

1. Rascon C, Meza I. Localization of sound sources in robotics: A review. *Robotics and Autonomous Systems*. 2017; 96:184–210. <https://doi.org/10.1016/j.robot.2017.07.011>
2. Corps RE, Knudsen B, Meyer AS. Overrated gaps: Inter-speaker gaps provide limited information about the timing of turns in conversation. *Cognition*. 2022; 223:105037. <https://doi.org/10.1016/j.cognition.2022.105037> PMID: 35123218
3. Stivers T, Enfield N, Brown P, Englert C, Hayashi M, Heinemann T, et al. Universals and cultural variation in turn-taking in conversation. *Proceedings of the National Academy of Sciences of the United States of America*. 2009; 106:10587–92. <https://doi.org/10.1073/pnas.0903616106> PMID: 19553212
4. Levinson SC, Torreira F. Timing in turn-taking and its implications for processing models of language. *Frontiers in Psychology*. 2015; 6. <https://doi.org/10.3389/fpsyg.2015.00731> PMID: 26124727
5. Levorato R, Pagello E. DOA Acoustic Source Localization in Mobile Robot Sensor Networks. In: 2015 IEEE International Conference on Autonomous Robot Systems and Competitions; 2015. p. 71–76.
6. Shi Z, Zhang L, Wang D. Audio ndash;Visual Sound Source Localization and Tracking Based on Mobile Robot for The Cocktail Party Problem. *Applied Sciences*. 2023; 13(10). <https://doi.org/10.3390/app13106056>
7. Knapp C, Carter G. The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustics, Speech, and Signal Processing*. 1976; 24(4):320–327. <https://doi.org/10.1109/TASSP.1976.1162830>
8. Risoud M, Hanson JN, Gauvrit F, Renard C, Lemesre PE, Bonne NX, et al. Sound source localization. *European Annals of Otorhinolaryngology, Head and Neck Diseases*. 2018; 135(4):259–264. <https://doi.org/10.1016/j.anorl.2018.04.009> PMID: 29731298

9. Santos JF, Senoussaoui M, Falk TH. An improved non-intrusive intelligibility metric for noisy and reverberant speech. In: 2014 14th International Workshop on Acoustic Signal Enhancement (IWAENC); 2014. p. 55–59.
10. Bergstra J, Komer B, Eliasmith C, Yamins D, Cox DD. Hyperopt: a Python library for model selection and hyperparameter optimization. *Computational Science Discovery*. 2015; 8(1):014008. <https://doi.org/10.1088/1749-4699/8/1/014008>
11. Heldner M, Edlund J. Pauses, gaps and overlaps in conversations. *Journal of Phonetics*. 2010; 38(4):555–568. <https://doi.org/10.1016/j.wocn.2010.08.002>
12. Barot P, MacDonald E, Mombaur K. Natural head and body orientation for humanoid robots during conversations with moving human partners through motion capture analysis. In: IEEE International Conference on Advanced Robotics and Social Impact; 2023.