*INTERSPEECH 2021: Show & Tell Contribution*
30 August – 3 September, 2021, Brno, Czechia

# Autonomous Robot for Measuring Room Impulse Responses

*Stefan Fragner*[1], *Tobias Topar*[1], *Maximilian Giller*[1], *Lukas Pfeifenberger*[2], *Franz Pernkopf*[1]

[1] Signal Processing and Speech Communication Laboratory, Graz University of Technology, Austria
[2] Evolve, Graz, Austria

{fragner, tobias.topar}@student.tugraz.at, maximilian.giller@sonible.com,
lukas.pfeifenberger@evolve.tech, pernkopf@tugraz.at

## Abstract

Far-field speech recognition for e.g. home automation or smart assistants has to cope with moving speakers in reverberant environments. Simulating stationary or even moving speakers in realistic environments enables to make speech processing technology more robust. This paper introduces an autonomous robot for recording a database of Room Impulse Responses (RIRs) at a high spatial resolution. This supports the creation of realistic simulation environments. These RIRs can be exploited to generate multi-channel speech mixtures of static or moving speakers for various applications.

**Index Terms**: room impulse response, measurement robot, reverberant speech data

## 1. Introduction

Obtaining real-world multi-channel recordings for speech databases is expensive and time-consuming. Therefore, multi-channel recordings are often artificially generated by convolving existing monaural speech recordings with simulated Room Impulse Responses (RIRs) from a so-called shoebox room [1] for stationary (not moving) speakers. Realistic scenarios such as home automation or smart assistants require to embed moving speakers in reverberant environments.

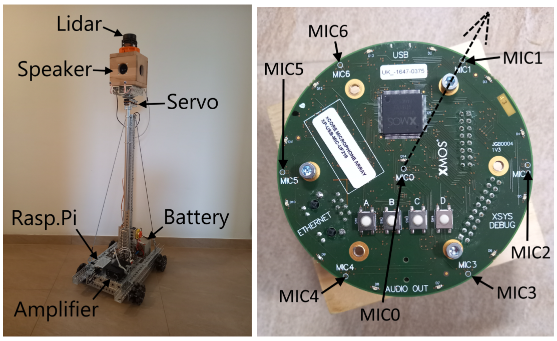
In this paper, we aim to support the generation of speech databases for these cases by recording multiple RIRs along a fine grid, spanning various reverberant environments (i.e. fully furnished office rooms). These RIRs can be used to simulate moving speakers by generating random trajectories on that grid, and quantize the trajectories along the grid points. For each matching grid point, the monaural speech recording can be convolved with the RIR at this grid point. Then, the spatialized recording can be compiled using the overlap-add method for each grid point.

To record the RIRs, we introduce an autonomous robot. The robot is able to navigate on a virtual grid at a pre-specified resolution by using a LiDAR system in combination with simultaneous localization and mapping (SLAM) based on the iterative closest points algorithm (ICP-SLAM) [2, 3]. To account for the looking direction of the speaker along its path, we record multiple RIRs at each grid point at specified angles. Overall, the recorded RIRs enable to generate data for speech separation in a reverberated and noisy environment including multiple channels and speakers.

The paper is organized as follows. Section 2 describes the robot while Section 3 introduce the features of the software measuring the RIRs. Finally, we conclude in Section 4.

## 2. Robot

The robot (see Figure 1a) consists of a solid aluminum frame with a pole in the center, which is strapped down by iron wires to ensure mechanical stability. On top of the pole is a servo motor which is used to rotate the platform above the servo. The platform can be rotated from 0° to 360°. A speaker cube is mounted on the platform above the servo. The speaker cube has four speakers but only the one in the front is connected and used. On top of the speaker is a 2D-LiDAR sensor. The LiDAR is attached to the speaker via a clicking mechanism so that it can easily be removed and mounted on top of the microphone.



(a) (b)

Figure 1: *(a) Robot; (b) Microphone array.*

Two geared motors, one on each side at the rear axle, are used to drive the robot. A raspberry pi is used to control the robot. In case of insufficient power, the measurement campaign can be continued without any data loss after charging.
Figure 1b shows the 7-channel microphone array [4]. The microphone array is attached to a base using some spacers. To enable the measurement of the location of the microphone array, in the room, the LiDAR can easily be detached from the top of the robot and mounted onto the microphone array. This process is discussed in more detail in Section 3.

Our recording setup consists of the 7-channel microphone array (XK-USB-MIC-UF216) and a 5W broadband loudspeaker (Visaton FR-58). To drive the loudspeaker from a Linux-based PC with ALSA [5], we use the sounddevice Python module [6], which can be used to play and record audio from a sound card. We use an exponential chirp with a duration of 5s, sweeping from 24 kHz down to 20 Hz as excitation signal to deconvolve the RIR [7]. To simulate a typical home assistant scenario, the microphone array is placed on a table, preferably in the middle of the room, while the robot moves on a virtual grid and records a set of RIRs at each grid cell at different angles $\theta$.

## 3. Software

For navigation and the gathering of data the robot is programmed in Python. The software provides a visualisation of

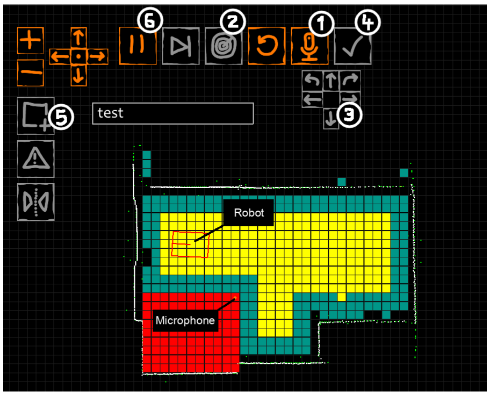


Figure 2: *Software interface.*

the recording process as well as the framework to ensure correct localisation and pathfinding of the robot. Localisation is achieved by ICP-SLAM [3, 2]. In particular a reference map of the room based on filtered LiDAR-data is created at the starting position of the robot. This map is partitioned into cells (or a grid) at a resolution provided by the user (see turquoise or yellow cells in Figure 2). In each of these cells a RIR measurement is performed.

Furthermore, the robot autonomously navigates to all cells. Upon arrival at a target position the speaker plays several chirps in different directions which are recorded by the microphone array. All the cells which have been visited by the robot are marked in yellow as shown in Figure 2. This process is repeated until every (reachable) cell is visited and RIR measurements have been performed. In case of unexpected incidents (e.g. robot gets pushed) and the robot is not able to match a scan to the previous ones, the user is prompted to align the marked scan to the reference map. This procedure is also used when pausing the recording, e.g. for switching the batteries. The software can be closed during this time, as long as the same room is used after a restart of the system.

As the LiDAR is only able to scan the room at a specific height (about 138 cm), objects like tables etc. might be invisible to the LiDAR. Therefore the user is able to mark areas within the room after the initial scan using the graphical interface (see red areas in Figure 2). The robot will avoid these areas. In addition mirrors and windows can be marked separately, as these would return incorrect distances of the LiDAR system. This procedure can also be used to mark areas where no RIR samples are needed. Marking can happen at any time while the automation is stopped.

**Localization of microphone array:** Measuring of the microphone position is also performed with the LiDAR. Therefore, the LiDAR can be flexibly mounted above the microphone array through a clicking mechanism. The scan from the microphone position is initialized using the interface (Microphone-Icon; number (1) in Figure 2). The room is scanned and the map is displayed. After that the LiDAR is mounted on the robot and the first manual scan can be obtained (Scan-Icon; number (2) in Figure 2). Next the user will be prompted to roughly align the microphone scan with the newly generated map (using the cursor buttons; number (3) in Figure 2). The fine alignment is performed automatically by pressing the Confirm Icon; number (4) in Figure 2). This procedure registers and saves the correct position of the microphone.

**Recording procedure:** The recording procedure is outlined in the following (see numbers in Fig 2)

1. Scan microphone position (Microphone-Icon (1))
2. Scan position of the robot (Scan-Icon (2))
3. Align microphone scan (Arrows (3) and Confirm (4))
4. Mark dangerous areas / mirrors / windows (Selection-Icon (5))
5. Start automation (Play/Pause-Icon (6))

**Synchronization:** Synchronization between the speaker and the microphone array is crucial. We propose to use the correlation of the signal directed towards the microphone. To facilitate this, it is required that a line-of-sight between the microphone and the speaker at any measurement position is available. In particular, we take one chirp for the initial position (facing the microphone) and one chirp for each of the angles where we want to measure. Then we concatenate these chirps by inserting pauses in between these chirps. The length of the pauses is determined by the time that is needed to rotate the speaker to the next angle position. Once the signal is concatenated, the time shift is determined by correlation of the first chirp in the recorded signal with the original chirp. Furthermore, we know the position of the speaker and the microphone array from the LiDAR measurement. Using both, the time shift of the chirp and the position of the speaker/microphone array, we are able to determine and correct the timing error between microphone array recording and speaker. As we know the length of the pauses in the concatenated audio this timing error correction can be exploited for all speaker angles.

## 4. Conclusion

We present an autonomous robot including a measurement and navigation software for recording a database of RIRs. In particular, all RIRs of a room at a pre-specified grid and angle resolution can be recorded without human intervention. Such RIR data supports the creation of realistic multi-channel recordings for far-field speech processing applications. Furthermore, we provide an approach for synchronization between the speaker and the microphone array based on correlation to account for timing errors.

In the future, we will record and provdie data of several rooms, including scripts for accessing this data. This enables to build realistic multi-channel speech data of static or moving speakers.

## 5. References

[1] R. Scheibler, E. Bezzam, and I. Dokmanic, “Pyroomacoustics: A python package for audio room simulations and array processing algorithms,” *CoRR*, vol. abs/1710.04196, 2017.

[2] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[3] “Pyicp-slam,” Website, visited on April 2nd 2021. [Online]. Available: https://github.com/gisbi-kim/PyICP-SLAM

[4] “Xk-usb-mic-uf216,” Website, available online at https://www.digikey.at/product-detail/en/xmos/XK-USB-MIC-UF216/880-1120-ND/6005986; visited on March 25th 2021.

[5] “Alsa-project,” Website, visited on February 19th 2020. [Online]. Available: https://alsa-project.org/wiki/Main_Page

[6] “python-sounddevice,” Website, visited on February 19th 2020. [Online]. Available: https://python-sounddevice.readthedocs.io/en/0.3.15/

[7] H. Kuttruff, *Room Acoustics*, 5th ed. London–New York: Spoon Press, 2009.