UNIVERSITY OF TARTU

Faculty of Science and Technology

Institute of Technology

Timur Nizamov

Audio System for the Social Humanoid Robot SemuBot

Bachelor's Thesis (12 ECTS)

Curriculum Science & Technology

Supervisor(s):

Associate professor of robotics engineering Karl Kruusamäe, PhD

Tartu 2024

Audio System for the Social Humanoid Robot SemuBot

Abstract

A huge and important part of a human companion robot is related to communication abilities. The humanoid robot should be able to hear the incoming speech, understand it and respond to it in the direction of a human that starts the interaction. To provide for such a solution, this thesis focuses on the design, testing and implementation of such a sound system for a real human companion robot.

The proposed sound system is composed of a microphone array board, a speaker system, a digital amplifier, and the Arduino MEGA for testing amplifier volume gain. Additionally, the ROS2 Humble is utilised for uniting the coded scripts and better communication with other parts of the robot.

The result of this work is shown on a real robot and together with other team members presented as a working prototype.

Keywords:

ROS2, Sound system, Humanoid robot, Microphone array board, Audio

CERCS: T120 Systems engineering, computer technology; T125 Automation, robotics, control engineering

Helisüsteem sotsiaalsele humanoidrobotile SemuBot

Lühikokkuvõte

Inimese kaaslase roboti tohutu ja oluline osa on seotud suhtlemisvõimetega. Humanoid-robot peaks suutma sissetulevat kõnet kuulda, sellest aru saada ja sellele reageerima inimese suunas, kes suhtlemist alustab. Sellise lahenduse pakkumiseks keskendub lõputöö sellise helisüsteemi projekteerimisele, testimisele ja rakendamisele tõelise inimese kaaslase roboti jaoks.

Kavandatav helisüsteem koosneb mikrofonide massiiviist, kõlarisüsteemist, digitaalsest võimendist ja Arduino MEGA-st, et testida võimendi helitugevust. Lisaks kasutatakse ROS2 Humble'i kodeeritud skriptide ühendamiseks ja paremaks suhtlemiseks roboti teiste osadega.

Selle töö tulemust näidatakse tegelikul robotil ja koos teiste meeskonnaliikmetega esitletakse töötava prototüübina.

Võtmesõnad:

ROS2, Helisüsteem, Humanoid robot, Mikrofonide massiiv, Audio

CERCS: T120 Süsteemitehnoloogia, arvutitehnoloogia; T125 Automatiseerimine, robootika, juhtimistehnika

TABLE OF CONTENTS

TERMS, ABBREVIATIONS AND NOTATIONS

ROS - Robot Operating System

DSP - Digital Signal Processing

AGC - Adaptive Gain Control

OSS - Open Source Software

AI - Artificial Intelligence

DOA - Direction Of Arrival

CAD - Computer-Aided Design

LCD - Liquid Crystal Display

ALSA - Advanced Linux Sound Architecture

1 INTRODUCTION

Recently, the impact of robotics technology has become drastically prevalent, changing and reforming various sectors from the food service industry [1] to the healthcare industry [2]. Robots themselves have now expanded their use even into our homes and workspaces [3]. Among various forms of robots, the section of social robots has also been quite prevalent in recent years, introducing us to humanoid robots that mimic human interaction with the usage of modern technology [4]. These robots are made to perform tasks and to be communicative and social with us.

The emerging topic of social robotics can be defined as fully or partially automated technologies that socially interact with humans, where robots take part in various services, where they perform tasks [5]. Unlike industry-related robots, social robots are engineered to interact and have a relationship with humans. Usually, they are equipped with multiple sophisticated sensors, screens, and artificial intelligence algorithms that allow them to sense the environment, recognise faces and gestures, and language models that enable them to hear and recognise the speech of different languages and most importantly form a correct response.

One crucial aspect of a social humanoid robot is its sound system, which is vital for communication. Through hardware devices, natural language processing, auditory feedback, and DSP filtering, social robots can have more meaningful conversations, have a better speech understanding, and express correct emotions. The design and quality of such a sound system are important and play a huge role in assessing a social robot's ability to maintain stable, meaningful, and efficient communication.

The goal of this thesis is to develop such a sound system for the humanoid social robot companion SemuBot that is intended for communication training in the therapy of children with special needs. The expected sound system consists of microphones for audio acquisition, speakers for playing the robot's answers, and an amplifier to manipulate the volume of the output signal.

2 LITERATURE REVIEW

2.1 Social robotics, humanoid robots, and open-source robotics

The field of social robotics is becoming a major topic of research together with the development of different technologies, such as artificial intelligence and human-computer interaction [6]. Social robots can be seen in educational, commercial, industrial, and home applications [6]. The interactive and dynamic features of such robots, together with their ability to hear, understand, and respond to human speech and emotions, make them a suitable candidate for the domain of social applications [7].

The humanoid robots, being from a subset of robotics, are made to resemble humans, which accounts for communication, appearance, and behaviour [8]. By imitating humans and having an anthropomorphic appearance, emotions, and gestures, humanoid robots can bring out more empathetic and trustworthy user interactions, enhancing user's emotional comfort [8], [9].

In the context of multiple fields where robots are implemented, the concept of open-source robotics has gained significant popularity due to its accessibility benefits [10]. The concept itself is related to OSS (Open Source Software). It utilises the same principles of source code distribution and includes licenses that allow developers to perform modifications in the software so that it would fit into a specific need, but in the field of robotics [11].

2.1.1 Social robots

Numerous social robots have been created and are available on the global market and each one of them is connected to a specific need. Nevertheless, they share similar features and functions.

The Misty II by Misty Robotics [12] is a personal robot that is aimed specifically at students, programmers, and entrepreneurs who are interested in robotics (Fig. 1). Robot has a 4K camera for image recognition, Occipital 3D depth sensor for three-dimensional room mapping, an LCD for expression displaying, a pair of speakers in the chest and three microphones [12].

Fig. 1. *The Misty II personal robot [12].*

Even though it is small in size, Misty's creators suggested that the robot can be used in healthcare (specifically elderly care), and building management (warehouse deployment), and there are also potential security applications [13].

Another prominent example of a social robot is the Vector 2.0 robot by Digital Dream Labs [14]. The robot utilises an HD camera for computer vision and navigation, a four-microphone array for directional hearing, touch sensors and an accelerometer for touch detection, and a processor [14]. In general, the robot was made to be a home social companion that is interactive and aware of the surroundings (Fig. 2).

Fig. 2. *The Vector 2.0 social companion* [14].

2.1.2 Humanoid robots

As we step into the humanoid subsection of social robots, we start to observe more anthropomorphic features and designs as most humanoid robots resemble human features and even share the same Degrees of Freedom as humans [15].

This can be seen in the robot Pepper (Fig. 3) by SoftBank Robotics that became a universal platform for business-to-consumer, business-to-academics, and business-to-developers areas, for example, Pepper was particularly deployed in schools and homes [16]. The humanoid has the capability of exhibiting body language, investigating and processing the surroundings, analysing human expressions, and identifying voice tones [16].

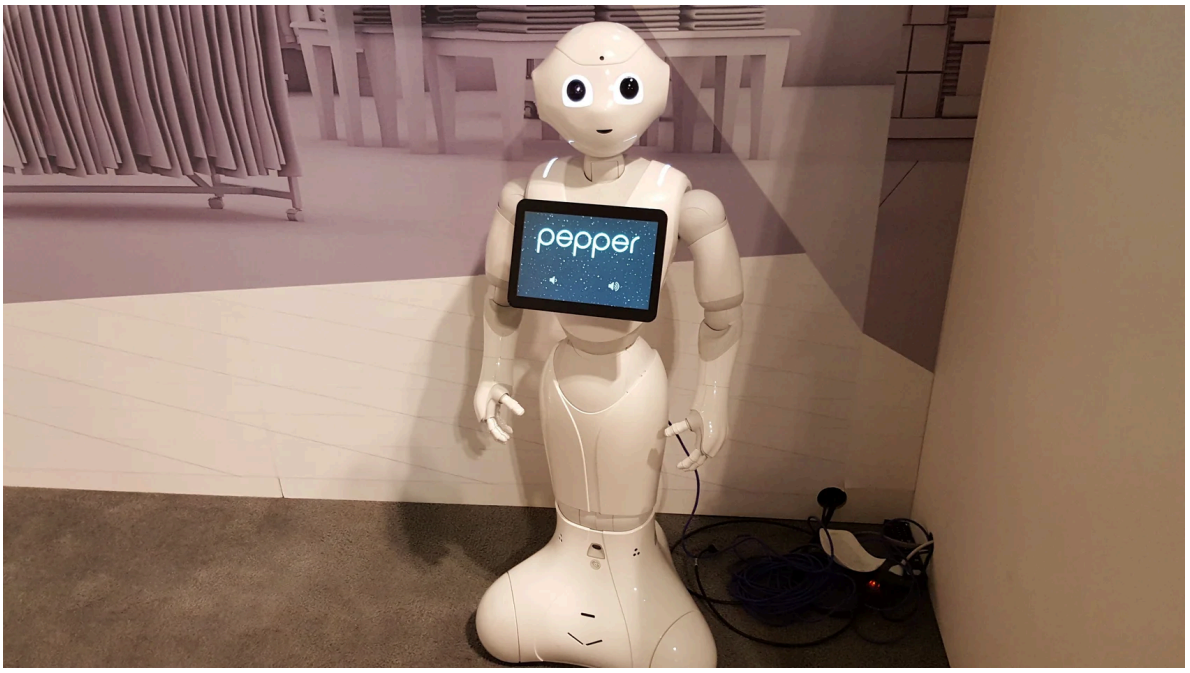
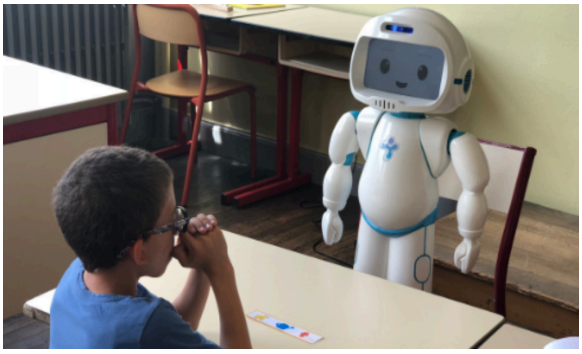


Fig. 3. *The Pepper humanoid robot [17].*

Another humanoid robot is the QTrobot by LuxAI which is used mainly in homes, schools, and for research purposes (Fig. 4a). Together with other sensors, the robot utilises a 4-microphone array for hearing and speakers, connected to a stereo amplifier, for the voice output [18].



(a) (b)

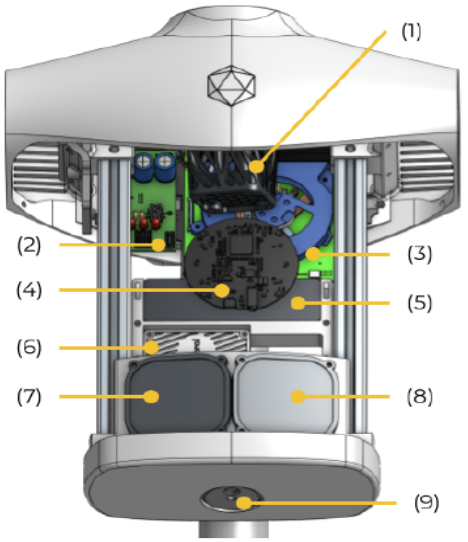
Fig. 4. *The QTrobot [18].*

2.1.3 Open-source robots

In the last years, the increasing number of projects in open-source robotics has been seen and those projects make it possible to benefit from other people's work and gain knowledge from shared robotics projects [19]. In this sphere, several notable robotics platforms have been created [20], [21], which provided other developers with useful information about both hardware and software. In this realm, robots are developed with open-source software and hardware together with publicly shared schematics and source code [22].

The Robotont omnidirectional robot is an open-source platform that has been successfully used for research and professional education, which supports the ROS software network (Section 2.4) that allows the user to learn about high-level robotic concepts [20]. All of the robot's hardware and software together with instructions are openly accessible and as a result, the list of its capabilities is constantly increasing and the robot itself has been a subject of research [20], [23], [24].

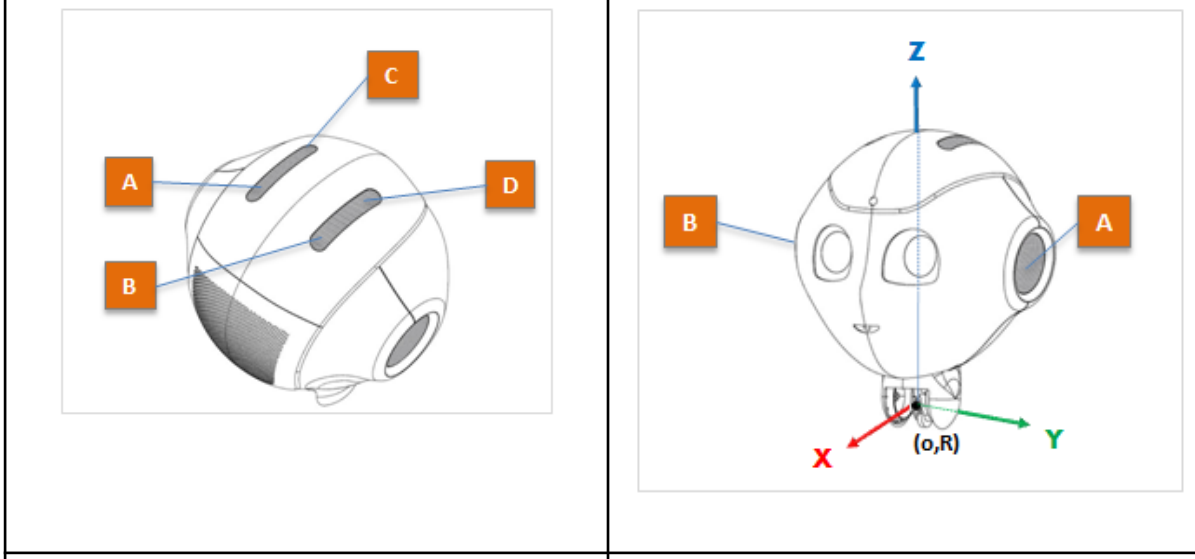
(a)



(b)

Fig. 5. *The Reachy robot [21].*

Another prominent example is the Reachy robot which represents an expressive fully open-source humanoid platform that utilises ROS2 Humble and can be used in research, healthcare, retail, and educational applications (Fig. 5a). The robot has several distinctive features, including VR teleoperation for remote control and state-of-art AI (Artificial Intelligence) frameworks. Similarly to the previous example, all its source code, mechanical specifications, CAD (Computer-Aided Design) models, and tutorials are openly distributed [21].



(a) (b)

Fig. 6. *Pepper's head [25].*

2.2 Audio Systems in Robots

To communicate and engage in meaningful conversations with humans, often in natural language, robots have to be equipped with an audio system, and as social robots need to capture human speech, process it, and play the generated output to communicate, a hardware solution has to be implemented. In terms of hardware, mainly a few notable components are used in social humanoid robots: a single microphone/multiple microphones in an array, speakers, and in some cases an amplifier.

The first example of an audio system that has these components is present in the QTrobot (Fig. 4b). It has double speakers located on the left and the right sides of the robot's head and they are connected to Raspberry PI powered by a 2.8W class D stereo amplifier [26]. For a microphone solution, the robot has an integrated digital microphone array in the head that is also connected to Raspberry Pi via a USB port [18]. The second example can be seen in the previously shown Pepper robot. Together with other sensors, the robot has loudspeakers, placed on the left (A) and the right (B) sides of the robotic head (Fig. 6b) and four microphones (A-D) on top of the head part for sound localization (Fig. 6a).

The last example of an audio system can be observed on the previously mentioned Reachy robot. The robot has a digital microphone array (label 4 in Fig. 5b) that is integrated not in the

head, but in the torso (Fig. 5b) and two 12W 4 Ohm speakers integrated in the same part (labels 7-8 in Fig. 5b) that are connected to the 12V audio amplifier [27]. In addition, there is a loudspeaker volume control button that is placed on the bottom of the torso (label 9 in Fig. 5b).

2.3 Digital Signal Processing (DSP) techniques

Digital signal processors are one of the most important features of modern audio equipment that are utilised in a wide variety of devices including headphones, speakers, and studio gear [28]. In principle, a digital signal processor is a specialised microprocessor that manipulates received digital signals and applies various algorithms and sound modifications to achieve a desired sound characteristic [29]. They are systems that perform mathematical functions and speed up the execution of audio-related algorithms. DSP can aid with sound quality by improving frequency and dynamic spectrum, removing or suppressing echo, and controlling noise and volume gain [28].

For instance, one approach can be seen in the Misty robot, where researchers picked a band-pass filter to increase the accuracy of DOA (Direction Of Arrival) estimation by cutting frequencies lower than 500 Hz and higher than 5000 Hz [30].

An example of a robot that used an audio system with DSP filters is the previously mentioned QTrobot and in addition, developers made a graphical tool to manipulate the parameters [31]. Of course, the filters can be also tuned to specific needs that depend on an environment or a specific use case scenario, however, in most cases, it is suggested to use a few parameters including automatic gain control, general manual gain regulation, and values for the amount of noise suppression [31].

2.4 Robot Operating System (ROS)

During the development of an open-source robot, the functionalities of all of the parts have to be combined physically and in terms of software, and here is where ROS can be useful. In general, ROS is an open-source framework for robots and a tool that offers node-based communication between different software modules, enabling data exchange. This is done either by making a node publish or/and subscribe to messages over a chosen topic or by implementing services that only provide data when they are called by a client [32].

ROS is multi-domain which means that it is ready for different applications, ranging from indoor to outdoor, from educational and home to industrial usage. It also provides us with tools and libraries for your specific robotics applications [33].

ROS2 is developed from scratch and has several upgrades including multiple platform support (Linux, Windows, and macOS) and processing of multiple nodes [34]. In addition, with ROS it is possible to have long-term support for its newer distributions (Foxy, Humble, Iron Irwini, etc) and stability with its core functionalities [35].

There are several robots where ROS2 was implemented including previously mentioned social and humanoid robots. ROS1's distributions are used in Pepper, QTrobot, and Robotont [16], [18], [20]. And Reachy [21], still being in active development, utilises the ROS2 Humble distribution.

3 THE AIMS OF THE THESIS

3.1 Creating an audio system for the SemuBot

My thesis work is part of a student project whose aim is to create a social humanoid robot SemuBot that can be potentially used as a communication trainer in the therapy of children with special needs. In the project, my goal is to create a sound system for such needs that would satisfy the requirements.

3.1.1 SemuBot requirements

- Can listen
- Can communicate (in Estonian)
- Should be able to move the head/eyes towards the speaker

3.1.2 System requirements

- ROS2 Humble
- Ubuntu 22.04 LTS

4 EXPERIMENTAL PART

To satisfy the requirements, the sound system was made, and for optimal positioning, the robot's head was chosen. The system can listen, move the eyes in the direction of a speaking human, and transfer the recorded audio for processing, after which the answer can be played back from speakers. The solution is divided into two major parts, software and hardware.

4.1 Hardware selection and analysis

By observing the hardware solutions of other social robots and looking at the overall characteristics the audio system parts were chosen. For the microphone, the Respeaker V2.0 mic array was chosen because of the number of useful DSP filters implemented inside and because of the previous usage in social robots [27], [31] and its proven functionality [31]. For the digital stereo amplifier, the Adafruit MAX9744 Class D was chosen for the amount of power that it can provide for each channel (20W) and the ability to digitally control the amplification. Lastly, the Visaton FR-87 4-ohm speakers were chosen as they nicely fit into the allowed amplifier power limit range (Fig. 7).

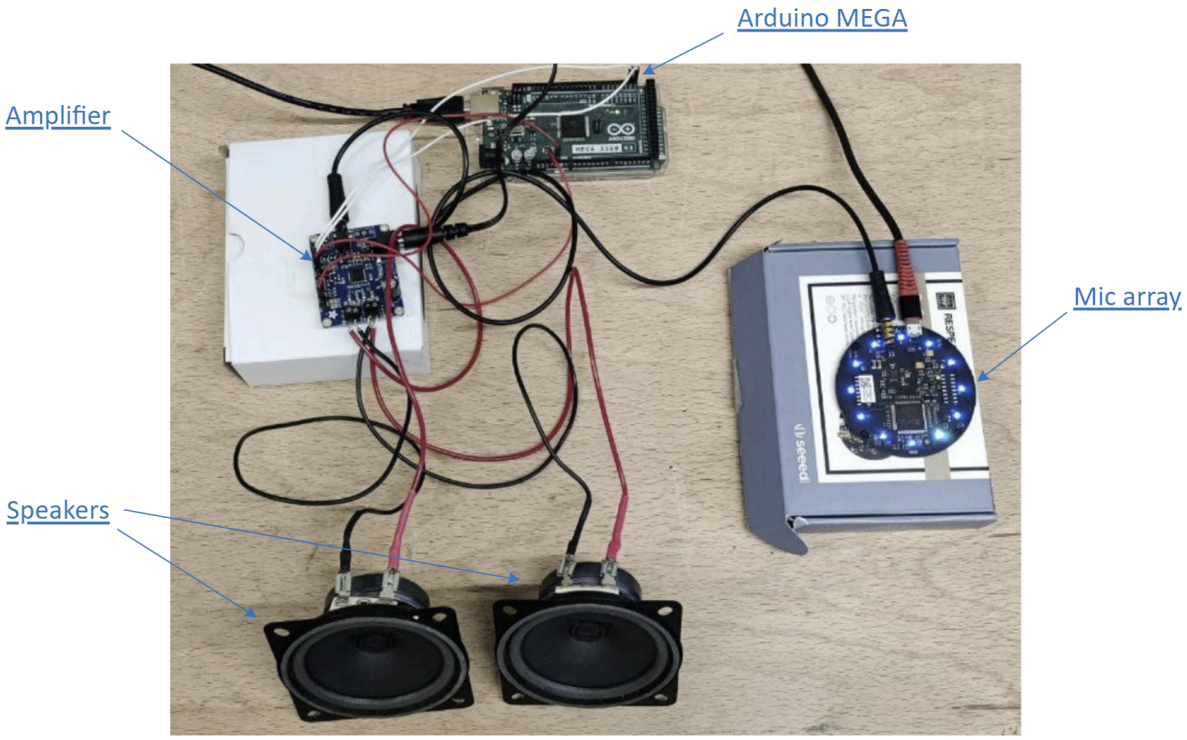


Fig. 7. *The constructed physical audio system*

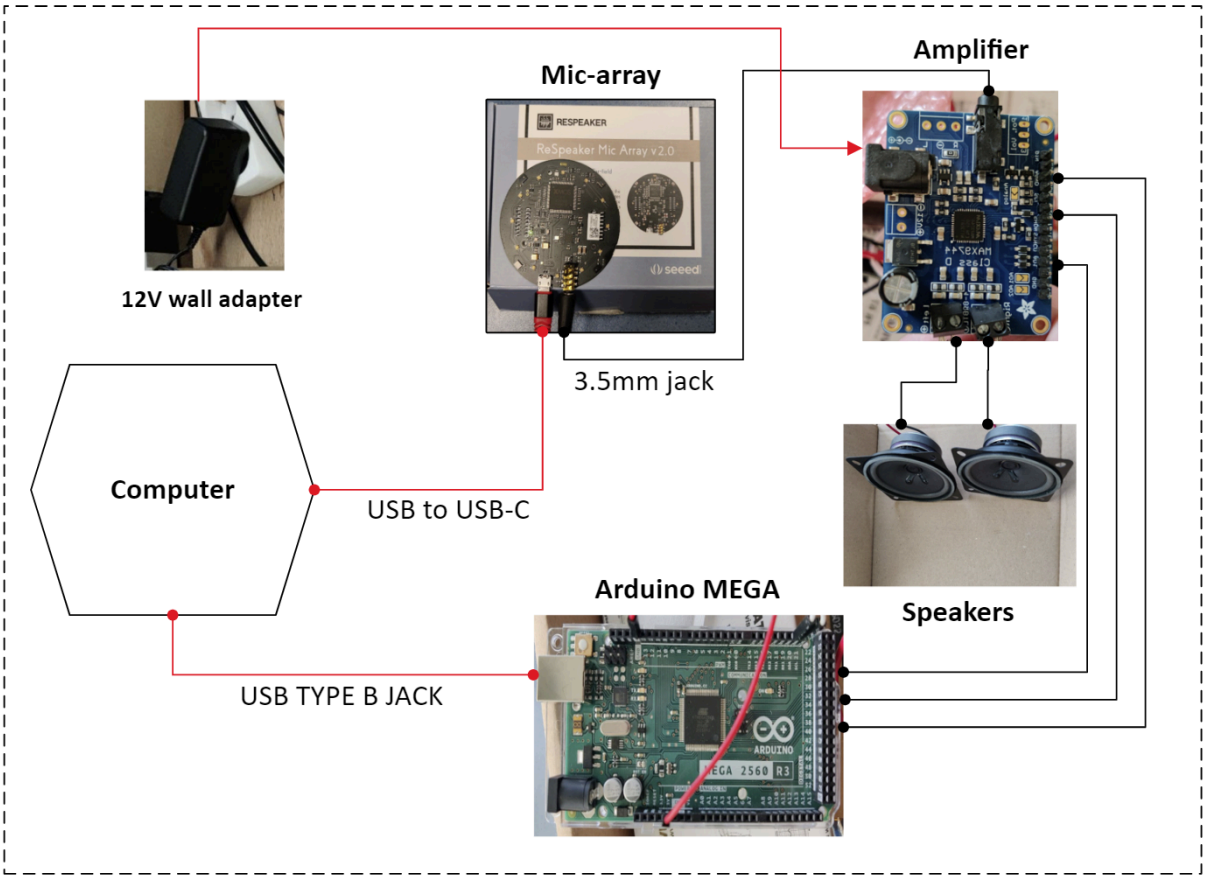


Fig. 8. *Audio system's connection diagram*

The mic array is connected with a USB-C to USB connector to the computer and with a 3.5mm jack to the amplifier. The amplifier is powered by the 12V wall adapter. Speakers are connected to the amplifier's terminal blocks by soldered electric wires. The pins of the amplifier were also soldered for digitally controlling the level of amplification. The Arduino MEGA is connected to the amplifier using 3 pins (GND, SDA, and SCL) and to the computer with a USB type B jack.

The system produced enough volume starting from the default amount of amplification, which is a good output. That means that the physical audio system (Fig. 8) is functional and can be later assembled into the robot.

4.2 Software implementation

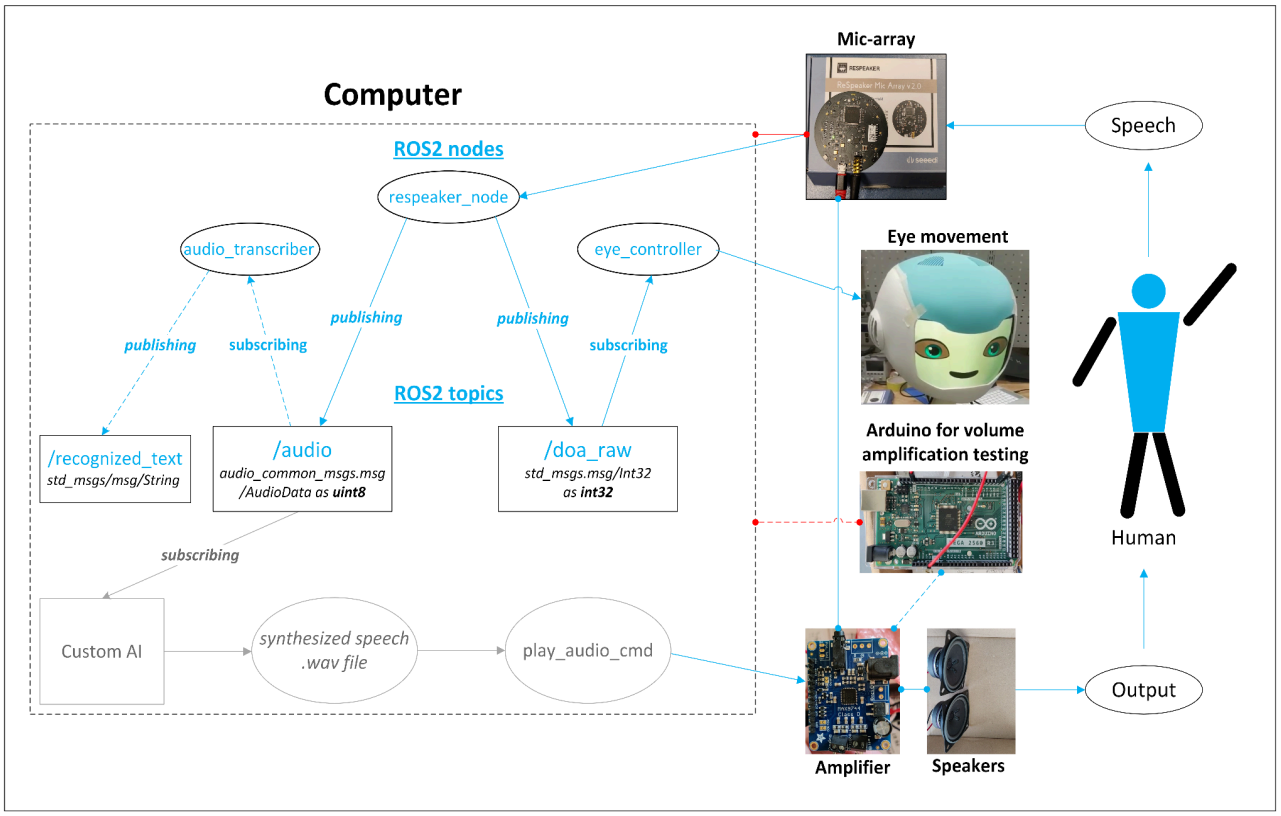


Fig. 9. *The general software solution made for the Semubot project consists of previously mentioned hardware and nodes that control the screen eye movement and send the DOA, Audio stream, and text data.*

The general schematic of the audio system is shown in Fig. 9, while the whole solution together with the setup instructions can be accessed through GitHub: https://github.com/SemuBot/nizamov-thesis-2024-semubot-audiosystem

The lines, namings, and headers that are coloured in light blue or red (hardware power connections) are in the scope of my thesis work, while the connections marked in grey are related to another thesis work [36] related to audio processing. Connections marked with dotted lines indicate the elements of software or hardware parts that were used for testing. The lines that have dots on both sides represent hardware connections. Light blue connections with an arrow show a general flow of data and its transmission and directionality.

Once everything is set up, the mic array is identified and the DSP parameters are automatically set, the *respeaker_node* node is initialised. The mic array takes the human speech in and can publish it to the */audio* topic in the node so that a constant audio

transmission is made. In addition, from the same *respeaker_node* node, the DOA data is published to the */doa_raw* topic, and at the same time a subscription is made on the *eye_controller* node that simultaneously proceeds with moving the eyes.

Then, as the audio stream is initialised, it is taken into the AI processing section, the output of which is, for now, a *.wav* file that can be played by the *play_audio_cmd* shell command. The *.wav* file is played through the system's speakers that are connected to the previously mentioned amplifier.

4.2.1 The respeaker_node

The *respeaker_node* data publishing node from the *respeaker_ros* package is written in Python programming language and was forked from the repository[1] of its developers and simplified due to occurring errors in one data transfer. To make the node function, another package *audio_common* [37] was utilised and built from the source. In addition, the source code[2] takes part in initializing the parameters of DSP filters (*parameters.py* and *interface.py*), suppression of the ALSA (Advanced Linux Sound Architecture) errors (*util.py*), and receiving the DOA from the mic array itself (*interface.py*). Scripts themselves are located within the same *respeaker_ros* directory together with the *respeaker_node.py* and imported as a package.

At first, the audio stream is initialised using Python's *pyaudio* package, and 2 publishers are activated for the DOA and the audio with the topics of */doa_raw* and */audio* encoded by *audio_common_msgs.msg/AudioData* and *std_msgs.msg/Int32* data types respectively. In a separate function, the DOA conversion into radians is done using Python's *math* module, and the data is published only if the angle is changed.

For testing purposes, the test *audio_test*[3] package was created, which contains the *audio_transcriber* publisher-subscriber node. Within the code, a subscription for the */audio* topic is initialised and a buffer with a duration of 8 seconds is made. In the *listener_callback()* function the received audio data is appended to the buffer which is then played through the system's speakers and processed through Google's *speech_recognition* library for Python [38] with a chosen parameter *'et-EE'* for transcribing Estonian speech. The result of the

[1] https://github.com/hcrlab/respeaker_ros
[2] https://github.com/SemuBot/respeaker_ros/tree/master/respeaker_ros
[3] https://github.com/SemuBot/audio_test

transcription, encoded by *std_msgs/msg/String data type*, is published over the topic */recognized_text* and printed in the terminal (Fig. 10). The buffer is cleared after publishing.

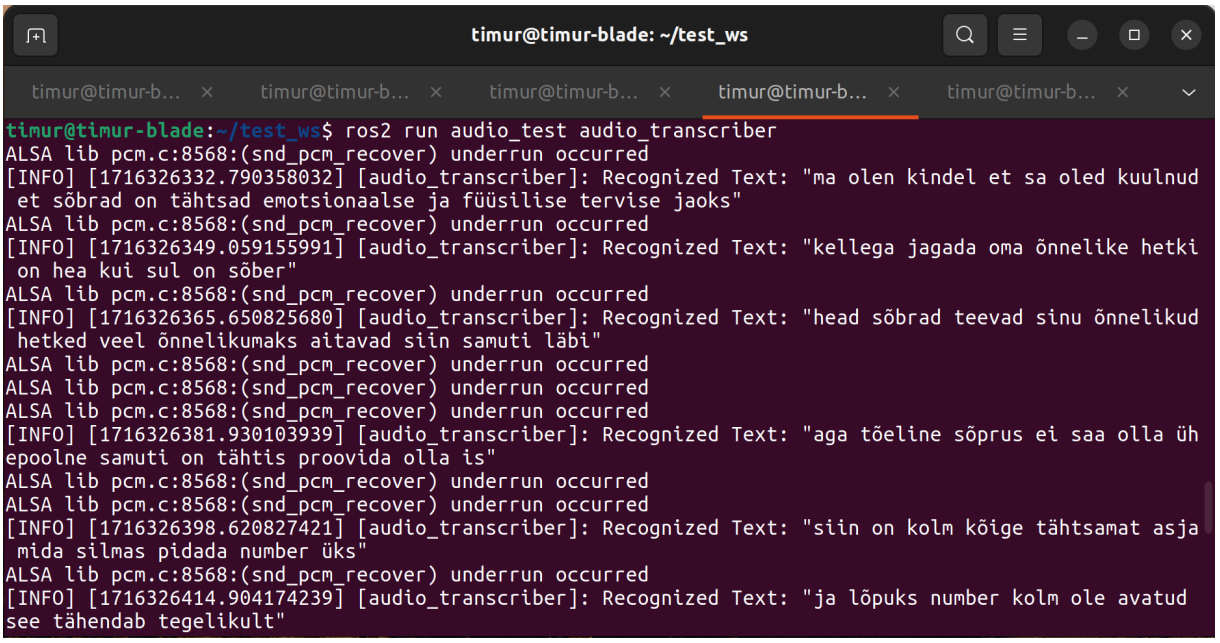


Fig. 10. *Transcribed Estonian text printed out in the terminal*

4.2.2 The eye_controller node

The *eye_controller* subscriber node accepts the incoming DOA data and uses 4 images (mouth, eyelids, outline, and eye)[4] in PNG format to visualise the eye movement. Python's *pygame* [39] library is utilised throughout the script for initialising the window, loading, scaling, and showing the images on the screen. After the *pygame* is initialised, window properties are set and images are loaded, a subscription for the DOA over the topic of */doa_raw* is made. Then, the data from the *respeaker_node* publisher node is received in the format of an angle ranging from -180 to 0 and from 0 to 180 degrees.

The images are rescaled and the eye images are moved on the screen in a fixed range of movement by manipulating the x-coordinates of the eye images. The x-coordinates are obtained by dividing the window's width by 2 and either subtracting or adding (left or right eye) an integer that can be defined as the distance between the x-coordinates between the eyes. The sketches of eyelids, outline, and mouth are shown on a screen statically so that their position on the screen does not change, while positions of the eye pictures are updated as the

[4] https://github.com/SemuBot/semubot_eyes/tree/main/images

DOA data is received. That is achieved in the function called *draw_eyes*. In the main() function, the displayed window is updated by calling the *draw_eyes* function in a while loop. In the same while loop, the *rclpy.spin_once()* function that uses the *rclpy* Python client library for ROS2 ensures that the *eye_controller* node can process incoming messages.

In addition, a non-blocking timeout of 0 was used to update the screen and at the same time be responsive to receiving data, allowing the main loop to continue executing without delay.

4.2.3 Characterisation

4.2.3.1 Methodology

Several tests were made to verify that the microphone is suitable for the needs of the project. It means that the mic array can provide steady, accurate DOA data for eye movement and an audio stream of such quality that would allow the Custom AI to transcribe it into text with minimal errors in transcription. The SNR (Signal-to-Noise Ratio) and Word Error Rate metrics were chosen because of their relevance to verify the quality of the input audio.

The measurements were taken with a 90-degree angle between two of the overall four microphones of the mic array (Fig. 11) in a large room with a low amount of noise. The mic array was connected to a computer. The recorded Estonian speech was played through a portable speaker (JBL GO2) for 13 seconds at 14 points from 0 to 7 metres from the mic array 3 times for each configuration and processed through Python's *speech_recognition* [38] library and the chosen SNR formula. In addition, the DOA was also recorded and averaged at these points. Later, the Word Error Rate (1) was calculated using Python's *werpy* [40] package by taking the correct words of a played sentence and the output from the speech recogniser. For visualisation, Python's *matplotlib.pyplot* [41] and *plotly.graph_objects* [42] modules were used.

$$WER = \frac{S + D + I}{N} \quad (1)$$

where:

S is the number of substitutions,

D is the number of deletions

I is the number of insertions

C is the number of correct words

N is the number of words in the reference (N = S + D + C)

The SNR formula (2) was taken from Matlab's documentation [43] and interpreted through Python language locally in the Jupyter Notebook.

$$SNR(x, y) = 10log_{10}\left(\frac{\sum_{i=1}^{n} x_i^2}{\sum_{j=1}^{n} y_i^2}\right) \quad (2)$$

where:

x is the input signal

y is the noise

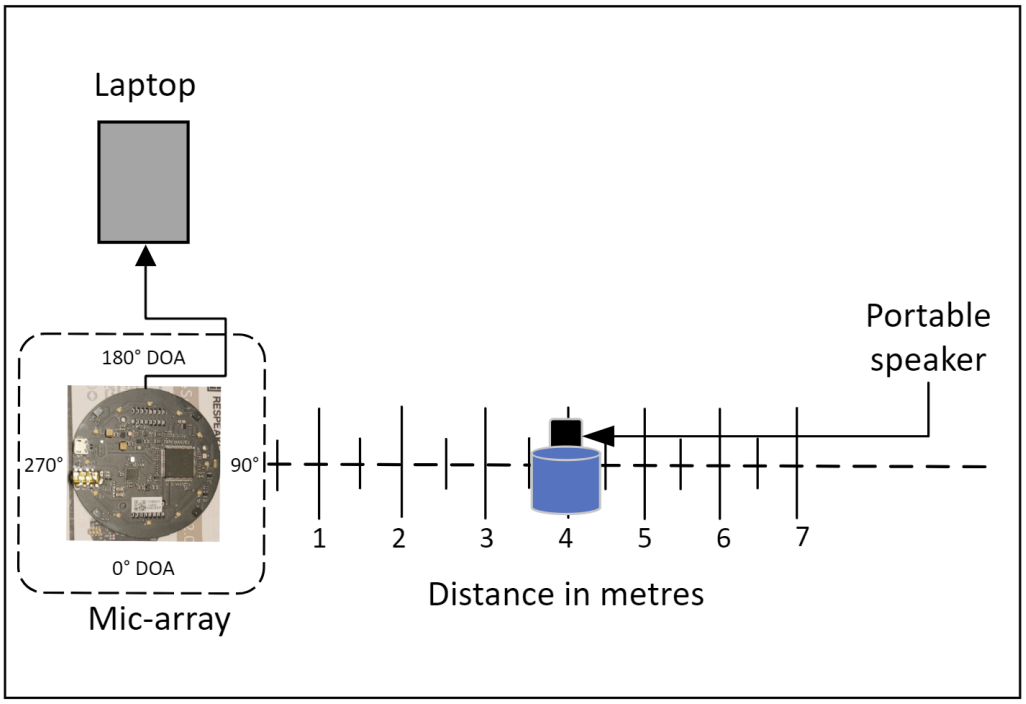


Fig. 11. *Geometric layout of the testing environment*

The tests were conducted in two configurations of the ReSpeaker mic array:

1) Config-default: the default configuration (is set every time the device is connected) with various internal DSP filters turned on.
2) Config-all-off: the configuration with no DSP filtering, the only parameter that is left is a constant gain

4.2.3.2 Results

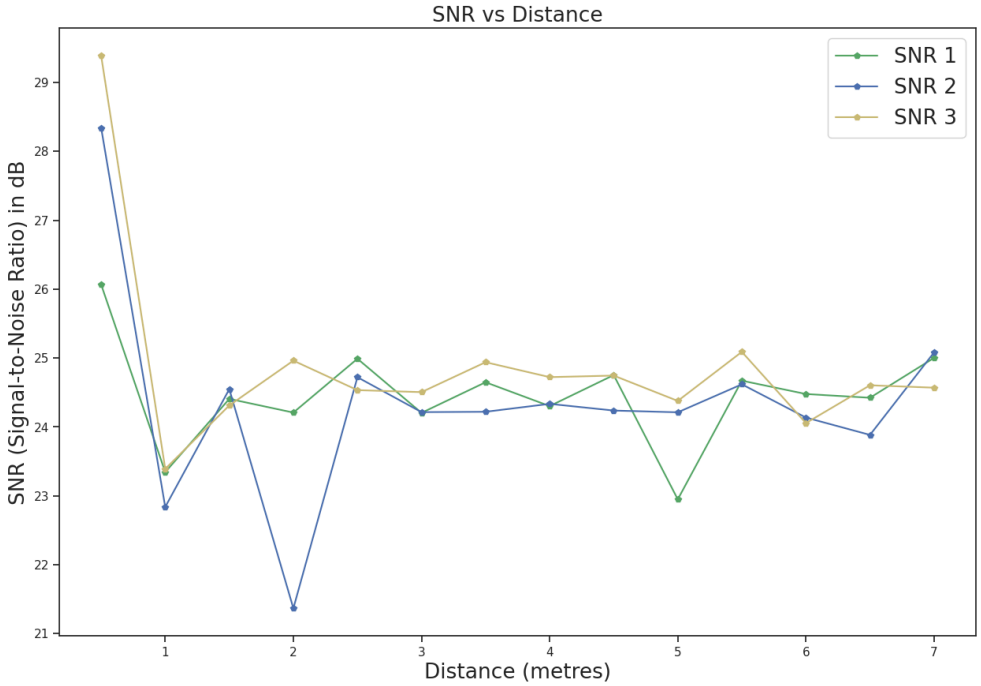


Fig. 12. *SNR vs Distance plot, Config-default*

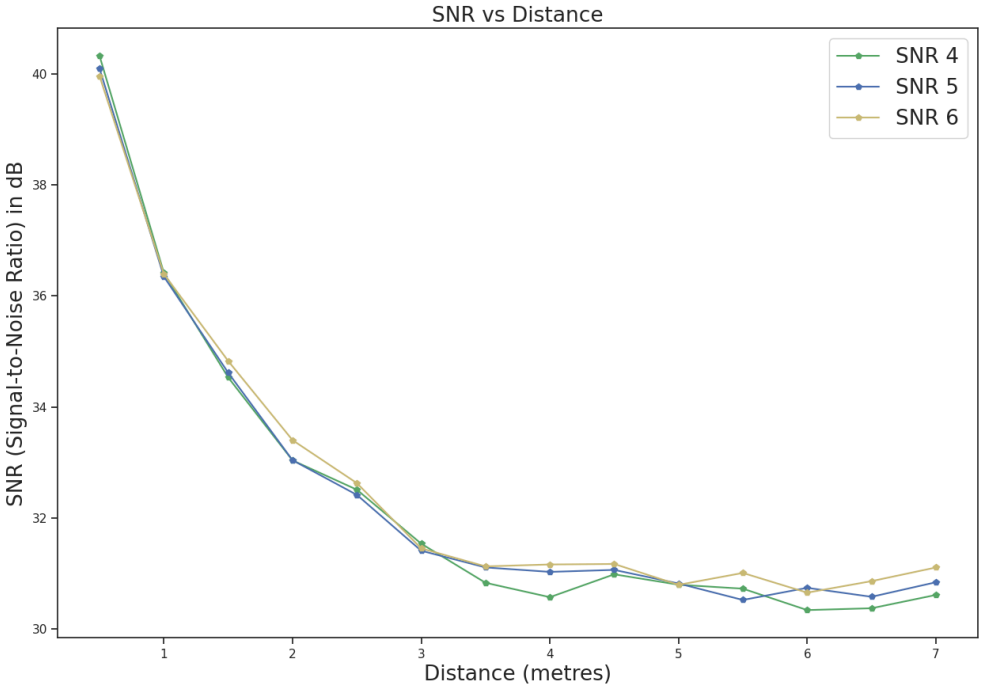


Fig. 13. *SNR vs Distance plot, Config-all-off*

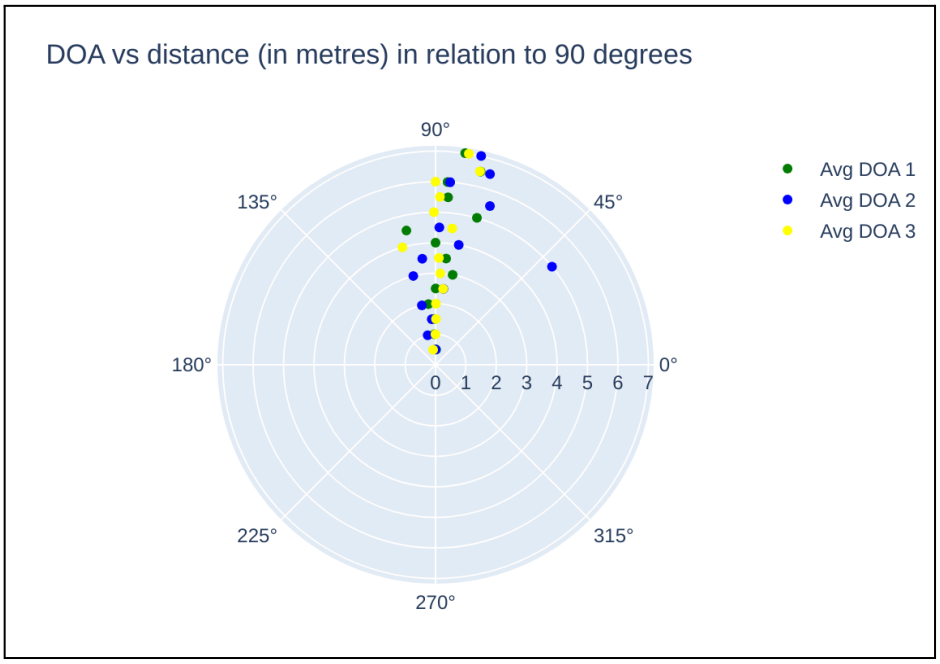


Fig. 14. *DOA (Direction Of Arrival) vs distance plot, Config-default*

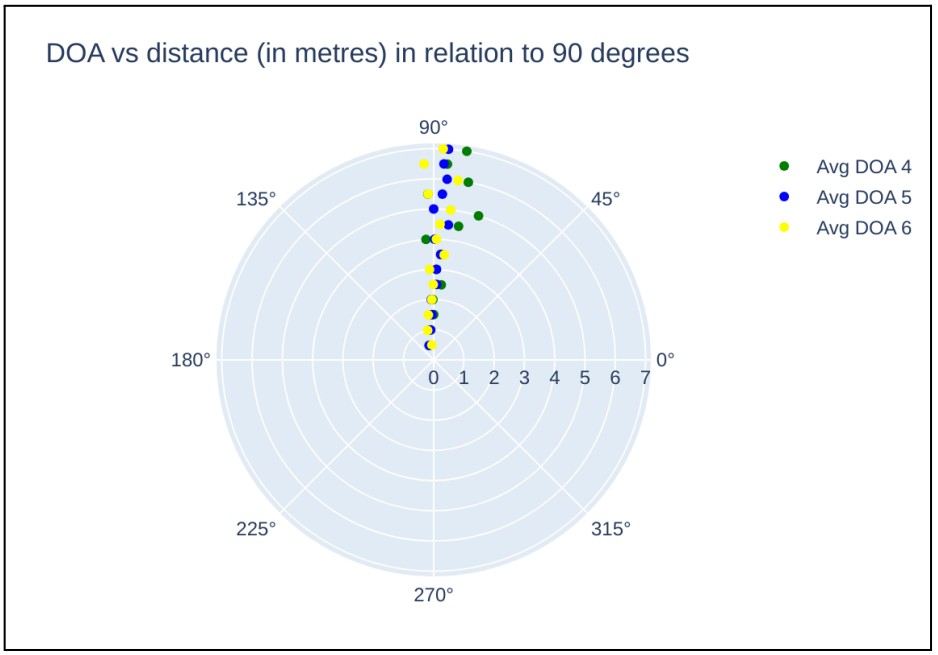


Fig. 15. *DOA (Direction Of Arrival) vs distance plot, Config-all-off*

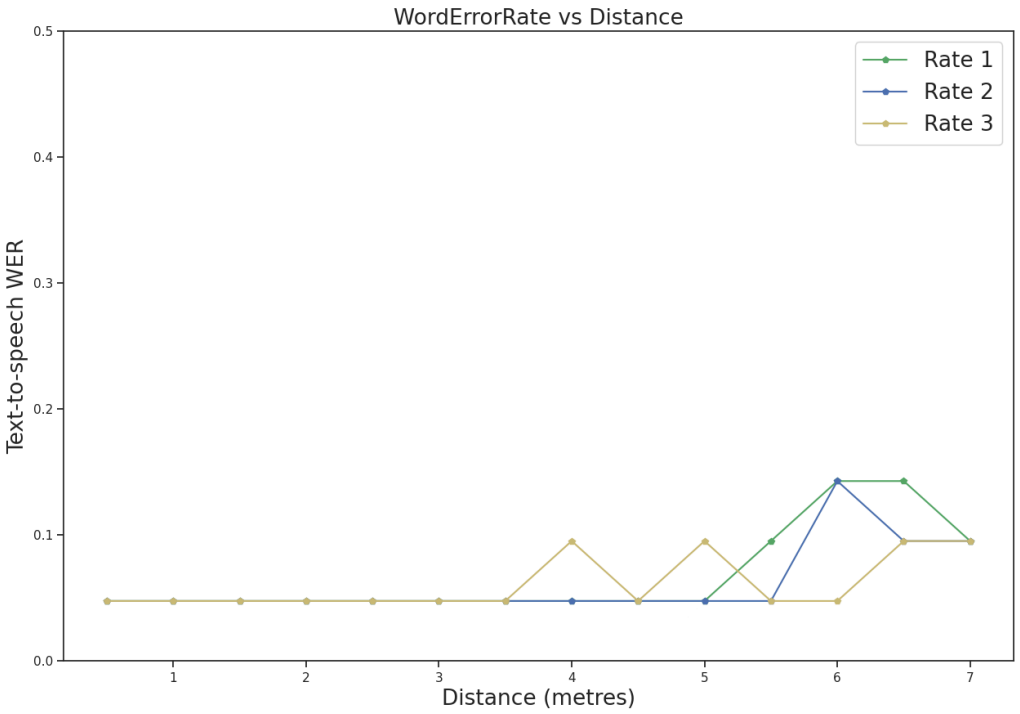


Fig. 16. *Word Error Rate vs distance plot, Config-default*

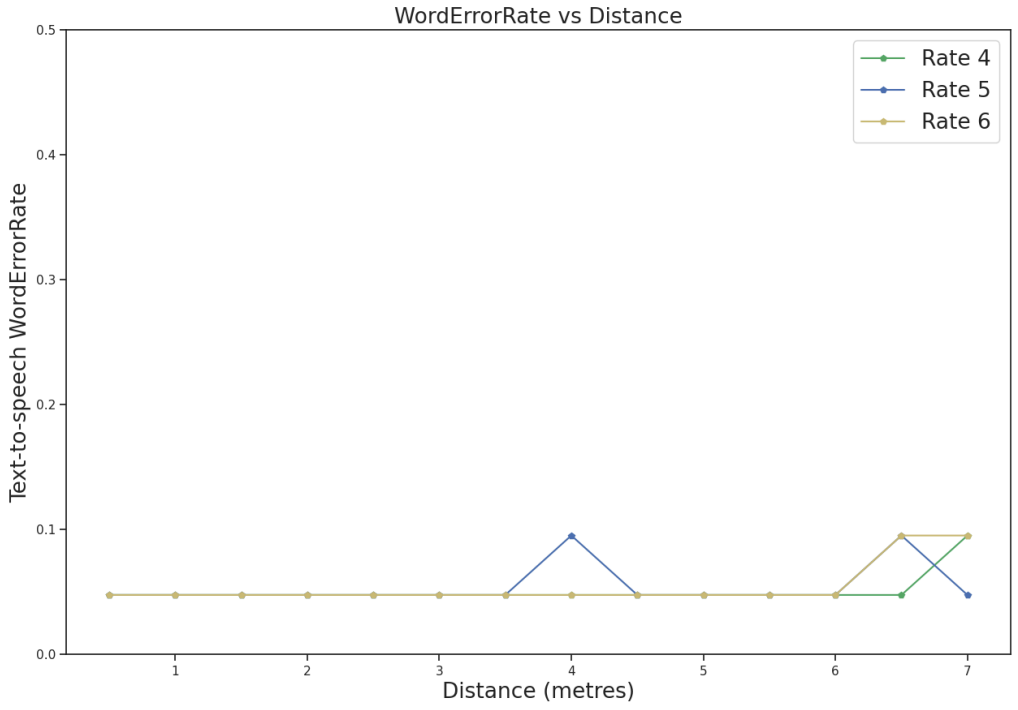


Fig. 17. *Word Error Rate vs distance plot, Config-all-off*

By looking at the first two graphs (Fig. 12-13) it can be seen that the SNR values differ from the start, which can be explained by the set constant gain that added overall volume.

The SNR value decrease curve is steeper in Fig. 13 in comparison to Fig. 12, for which the AGC (Automatic Gain Control) feature was turned on, resulting in a bit more stable SNR values. The DOA values are more centred to a 90-degree angle for the Config-all-off in comparison to the Config-default (Fig. 14–15). Lastly, the Word Error Rate (Fig. 16-17) is less for the Config-all-off in contrast to the Config-default. In addition, it can be seen that in both configurations a stable error is present at various distances. The DOA with 90 degrees is relatively stable for both configurations as the deviations are not big and partially caused by speaker positioning in relation to the mic array.

4.2.3.3 Discussion and conclusions

The Word Error Rate values (Fig. 16-17) ranging from approximately 5% to 10% (and up to 15% for the Config-default at 6-7 metres) can be considered good when compared to standards by industry leaders such as Microsoft [44]. Although these values mostly assess the quality of the language model, important tendencies can be observed, which can indicate that the quality of the incoming audio starts to decrease. Based on these tendencies, the preferred microphone usage distance can be derived. Even though the tests were made only in the noiseless environment and the numbers are approximate, good results can be seen, which means that the mic array can be used for future applications and more practical testing.

From the obtained Distance information, it can be concluded that the preferred range of usage would equal approximately 4 metres as the WER metric starts to decrease together with a small decrease in the DOA estimation accuracy. The decrease is not critical so the mic array can be used for the eye movement requirement.

4.3 Assembly of the audio system on the robot

Fig. 18. *The hardware assembly on the SemuBot robot*

The hardware was integrated on top of the robot's head and was assembled by one of the SemuBot team members.

4.4 Discussion

The provided hardware and software solutions for the robot's sound system fully satisfy the requirements that were previously listed (Section 3). The software part is run on the Ubuntu 22.04 system and implemented in ROS2 Humble. After research, the hardware was chosen (Section 4.1) and tested (Section 4.2.3). After quality verification, the hardware solution was assembled (Section 4.3). In software, the publisher-subscriber node logic was written in Python language and 3 types of data can overall be transmitted:

1) DOA, which controls the eye movement of the robot's face.
2) Audio stream that is sent for AI audio processing.
3) Text from audio transcription

The software solution is provided in the form of three ROS2 packages that are available on GitHub together with written documentation:

https://github.com/SemuBot/nizamov-thesis-2024-semubot-audiosystem

The general depiction of how the system functions is presented at the start of Section 4.2.

In addition, after installation guidelines for created packages were written, my work was successfully integrated with the Custom AI prepared by another SemuBot member [36]. As a result of this cooperation, after running all the nodes together with AI processing and speaking to the robot, a response can be generated. Visuals, including two videos with response generation and other pictures, are available on Github[5].

4.4.1 Limitations and challenges

During the experimental part of my thesis I faced several challenges and a more general limitation during the search for the hardware. Firstly, as ROS2 Humble was used, at the time of making the packages and writing the code the package *audio_common* was not fully ported, which caused some issues due to unsolved problems within its structure. That eventually led to errors on the project's main computer when following the package importing guidelines that were written for one of the created ROS2 packages, even though the same error could have been easily fixed on my laptop. That was quickly solved by not building the whole *audio_common* package, but only its three subpackages which excluded *sound_play* and *audio_common* (a subpackage). Eventually, a different method of playing the audio was used which is mentioned at the start of Section 4.2.

Secondly, after trying out the *respeaker_ros* package that was made by developers on GitHub, the quality of the audio stream that was accepted on my subscriber node was incredibly bad and the overall signal seemed to be distorted. After making a fork from the repository[6] and simplifying the code the audio stream started to be comprehensible.

Lastly, the limitation is related to the chosen mic array, as it is simply discontinued, which also implies that there is no active support for it. In the future, a newer microphone system can be implemented. If choosing between other available newer microphone arrays, a potential solution should be in production and with active support from its developers.

[5] https://github.com/SemuBot/nizamov-thesis-2024-semubot-audiosystem/tree/main/Visuals
[6] https://github.com/hcrlab/respeaker_ros

4.4.2 Future work

Firstly, the future work related to the SemuBot project will mainly consist of improvements to the created packages that imply changes to the code structure for a better presentation. In addition, an investigation regarding the original *respeaker_ros* forked ROS2 package will be made to find the causing error.

Secondly, as the tests were made only in low-noise conditions, more practical testing has to be conducted to see how the mic-array performs together with other parts of the robot and to tune the parameters of the DSP filters to achieve the optimal configuration.

5 CONCLUSION

The audio system which relies on human speech for communication was successfully made. It is composed of hardware and software parts that were implemented into the SemuBot social humanoid robot. The hardware parts were proved to be suitable, including the microphone which was tested in two different configurations of DSP filters. The software was made using ROS2 and was documented. The working prototype of the system was collaboratively presented on a real robot, which as a result can look at the speaker, listen, and give a response.

ACKNOWLEDGEMENTS

First, I would like to thank my supervisor Karl Kruusamäe for the guidance and valuable suggestions given throughout my thesis work.

Second, I would like to thank the team members of the SemuBot project who provided additional support throughout the development of my solution.

Additionally, Grammarly [45] was utilised for proofreading and correcting grammatical mistakes. ChatGPT-3.5 [46] provided help in improving the Introduction part of this thesis.

REFERENCES

[1] ‘Robots in the Food Service Industry | RoboticsTomorrow’. Accessed: Feb. 21, 2024. [Online]. Available: https://roboticstomorrow.com/article/2021/06/robots-in-the-food-service-industry/17029

[2] R. in the medical field are transforming how surgeries are performed *et al.*, ‘Robotics in Healthcare: The Future of Robots in Medicine’, Intel. Accessed: Feb. 21, 2024. [Online]. Available: https://www.intel.com/content/www/us/en/healthcare-it/robotics-in-healthcare.html

[3] R. Heilweil, ‘Robot, I’m home!’, Vox. Accessed: Feb. 21, 2024. [Online]. Available: https://www.vox.com/recode/23280840/smart-home-automation-robots-chores

[4] D. Mishra, G. A. Romero, A. Pande, B. Nachenahalli Bhuthegowda, D. Chaskopoulos, and B. Shrestha, ‘An Exploration of the Pepper Robot’s Capabilities: Unveiling Its Potential’, *Appl. Sci.*, vol. 14, no. 1, Art. no. 1, Jan. 2024, doi: 10.3390/app14010110.

[5] M. Čaić, D. Mahr, and G. Oderkerken-Schröder, ‘Value of social robots in services: social cognition perspective’, *J. Serv. Mark.*, vol. 33, no. 4, pp. 463–478, Jan. 2019, doi: 10.1108/JSM-02-2018-0080.

[6] K. Youssef, S. Said, S. Alkork, and T. Beyrouthy, ‘A Survey on Recent Advances in Social Robotics’, *Robotics*, vol. 11, no. 4, Art. no. 4, Aug. 2022, doi: 10.3390/robotics11040075.

[7] T. J. Prescott and J. M. Robillard, ‘Are friends electric? The benefits and risks of human-robot relationships’, *iScience*, vol. 24, no. 1, p. 101993, Dec. 2020, doi: 10.1016/j.isci.2020.101993.

[8] J. Fox and A. Gambino, ‘Relationship Development with Humanoid Social Robots: Applying Interpersonal Theories to Human–Robot Interaction’, *Cyberpsychology Behav. Soc. Netw.*, vol. 24, no. 5, pp. 294–299, May 2021, doi: 10.1089/cyber.2020.0181.

[9] L. Vianello *et al.*, ‘Human-Humanoid Interaction and Cooperation: a Review’, *Curr. Robot. Rep.*, vol. 2, no. 4, pp. 441–454, Dec. 2021, doi: 10.1007/s43154-021-00068-z.

[10] kgay, ‘Benefits of an open-source language for robots’, Plant Engineering. Accessed: Feb. 28, 2024. [Online]. Available: https://www.plantengineering.com/articles/benefits-of-an-open-source-language-for-robots/

[11] R. Davey, ‘What is Open Source Robotics?’, AZoRobotics. Accessed: Feb. 27, 2024. [Online]. Available: https://www.azorobotics.com/Article.aspx?ArticleID=427

[12] ‘Misty Robotics rolls out accessible, affordable personal robot’, New Atlas. Accessed: Mar. 01, 2024. [Online]. Available: https://newatlas.com/misty-robotics-personal-robot/54463/

[13] S. Colaner, ‘Meet Misty, an extensible robot empowered by Microsoft and Qualcomm technologies’, VentureBeat. Accessed: Mar. 01, 2024. [Online]. Available: https://venturebeat.com/ai/misty-an-extensible-robot-empowered-by-microsoft-and-qualcomm-technologies/

[14] ‘Vector 2.0 AI Robot Companion, Smart Home Robot with Alexa Built-in’, Digital Dream Labs. Accessed: Mar. 01, 2024. [Online]. Available: https://ddlbots.com/products/vector-robot

[15] M. Mesbahi, ‘Design Considerations for Humanoid Robots’. Accessed: Mar. 01, 2024. [Online]. Available: https://www.wevolver.com/article/design-considerations-for-humanoid-robots

[16] A. K. Pandey and R. Gelin, ‘A Mass-Produced Sociable Humanoid Robot: Pepper: The First Machine of Its Kind’, *IEEE Robot. Autom. Mag.*, vol. PP, pp. 1–1, Jul. 2018, doi: 10.1109/MRA.2018.2833157.

[17] M. Nardine, ‘Pepper Robot The Future of Retail’. Accessed: Mar. 02, 2024. [Online]. Available: https://www.robotlab.com/blog/pepper-robot-the-future-of-retail

[18] ‘LuxAI - Award winning social robots for autism and special needs education’, LuxAI S.A. Accessed: Mar. 02, 2024. [Online]. Available: https://luxai.com/

[19] V. V. Patel, M. V. Liarokapis, and A. M. Dollar, ‘Open Robot Hardware: Progress, Benefits, Challenges, and Best Practices’, *IEEE Robot. Autom. Mag.*, vol. 30, no. 3, pp. 123–148, Sep. 2023, doi: 10.1109/MRA.2022.3225725.

[20] R. Raudmäe *et al.*, ‘ROBOTONT – Open-source and ROS-supported omnidirectional mobile robot for education and research’, *HardwareX*, vol. 14, p. e00436, Jun. 2023, doi: 10.1016/j.ohx.2023.e00436.

[21] ‘Discover Reachy, a robotic platform based on AI – Reachy by Pollen Robotics, an open source programmable humanoid robot’. Accessed: Mar. 11, 2024. [Online]. Available: https://www.pollen-robotics.com/reachy/

[22] ‘Open-source robotics’, *Wikipedia*. Feb. 20, 2024. Accessed: Mar. 02, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Open-source_robotics&oldid=1209180120

[23] A. Jakovleva, ‘Roboquiz - an interactive human-robot game’, 2022, Accessed: Mar. 15, 2024. [Online]. Available: http://hdl.handle.net/10062/83037

[24] K. Sein, ‘Eestikeelse kõnesünteesi võimaldamine robootika arendusplatvormil ROS’,

2020, Accessed: Mar. 15, 2024. [Online]. Available: http://hdl.handle.net/10062/72102

[25] ‘Pepper - Technical overview — Aldebaran 2.0.6.8 documentation’. Accessed: Mar. 02, 2024. [Online]. Available: http://doc.aldebaran.com/2-0/family/juliette_technical/index_juliette.html

[26] ‘QTrobot Sound and Speech | QTrobot Documentation’. Accessed: Mar. 18, 2024. [Online]. Available: https://docs.luxai.com/docs/modules/speakers

[27] ‘Torso specifications’, Reachy 2023. Accessed: Mar. 18, 2024. [Online]. Available: https://docs.pollen-robotics.com/advanced/specifications/torso-specs/

[28] ‘What Is DSP In Audio? - Audiosolace’. Accessed: Mar. 18, 2024. [Online]. Available: https://audiosolace.com/what-is-dsp-in-audio/

[29] S. Welch, ‘DSP Benefits in Sound Systems: A Complete Guide’, Audio Intensity. Accessed: Mar. 19, 2024. [Online]. Available: https://audiointensity.com/blogs/dsp-amplifiers/dsp-benefits-in-sound-systems-a-complete-guide

[30] I. Ciuffreda, G. Battista, S. Casaccia, and G. M. Revel, ‘People detection measurement setup based on a DOA approach implemented on a sensorised social robot’, *Meas. Sens.*, vol. 25, p. 100649, Feb. 2023, doi: 10.1016/j.measen.2022.100649.

[31] ‘QTrobot Audio processing and Microphone | QTrobot Documentation’. Accessed: Mar. 19, 2024. [Online]. Available: https://docs.luxai.com/docs/modules/microphone

[32] ‘Documentation - ROS Wiki’. Accessed: Mar. 16, 2024. [Online]. Available: https://wiki.ros.org/

[33] ‘ROS: Why ROS?’ Accessed: Mar. 16, 2024. [Online]. Available: https://www.ros.org/blog/why-ros/

[34] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, ‘Robot Operating System 2: Design, architecture, and uses in the wild’, *Sci. Robot.*, vol. 7, no. 66, p. eabm6074, May 2022, doi: 10.1126/scirobotics.abm6074.

[35] ed, ‘ROS1 vs ROS2, Practical Overview For ROS Developers’, The Robotics Back-End. Accessed: Mar. 16, 2024. [Online]. Available: https://roboticsbackend.com/ros1-vs-ros2-practical-overview/

[36] ‘SemuBot/Unn-Thesis-2024-Semubot-SpeechSystem’. SemuBot, May 15, 2024. Accessed: May 21, 2024. [Online]. Available: https://github.com/SemuBot/Unn-Thesis-2024-Semubot-SpeechSystem

[37] ‘ros-drivers/audio_common: Common code for working with audio in ROS’. Accessed: May 20, 2024. [Online]. Available: https://github.com/ros-drivers/audio_common

[38] ‘SpeechRecognition: Library for performing speech recognition, with support for several engines and APIs, online and offline.’ Accessed: May 20, 2024. [MacOS :: MacOS X, Microsoft :: Windows, Other OS, POSIX :: Linux]. Available: https://github.com/Uberi/speech_recognition#readme

[39] ‘pygame: Python Game Development’. Accessed: May 21, 2024. [MacOS, Microsoft :: Windows, POSIX, Unix]. Available: https://www.pygame.org

[40] ‘werpy: A powerful yet lightweight Python package to calculate and analyze the Word Error Rate (WER).’

[41] ‘matplotlib.pyplot — Matplotlib 3.9.0 documentation’. Accessed: May 21, 2024. [Online]. Available: https://matplotlib.org/stable/api/pyplot_summary.html

[42] ‘Graph’. Accessed: May 21, 2024. [Online]. Available: https://plotly.com/python/graph-objects/

[43] ‘Signal-to-noise ratio - MATLAB snr - MathWorks Nordic’. Accessed: Apr. 26, 2024. [Online]. Available: https://se.mathworks.com/help/signal/ref/snr.html

[44] eric-urban, ‘Test accuracy of a custom speech model - Speech service - Azure AI services’. Accessed: Apr. 26, 2024. [Online]. Available: https://learn.microsoft.com/en-us/azure/ai-services/speech-service/how-to-custom-speech-evaluate-data

[45] ‘Grammarly: Free AI Writing Assistance’. Accessed: May 20, 2024. [Online]. Available: https://www.grammarly.com/

[46] ‘ChatGPT’. Accessed: May 20, 2024. [Online]. Available: https://openai.com/chatgpt/

APPENDIX

The created software solution together with hardware pictures, video demonstrations, and testing scripts related to my thesis work can be assessed in the following repository:

https://github.com/SemuBot/nizamov-thesis-2024-semubot-audiosystem

NON-EXCLUSIVE LICENCE TO REPRODUCE THE THESIS AND MAKE THE THESIS PUBLIC

I, Timur Nizamov,

1. grant the University of Tartu a free permit (non-exclusive licence) to

reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright, my thesis

Audio System for the Social Humanoid Robot SemuBot

supervised by Karl Kruusamäe.

2. I grant the University of Tartu a permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in points 1 and 2.

4. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Timur Nizamov

22/05/2024