Baltic J. Modern Computing, Vol. 12 (2024), No. 1, 82-96
https://doi.org/10.22364/bjmc.2024.12.1.05

Human – Robot Interaction Communication Control System Using Lithuanian Language

Linas AIDOKAS

Institute of Data Science and Digital Technologies, Vilnius University, Akademijos st. 4, LT-08663, Lithuania

linas.aidokas@mif.vu.lt

ORCID 0000-0001-6375-887X

Abstract. The field of human-computer interaction has progressed very rapidly over the last few decades, creating another area of research - human-robot interaction (HRI). This paper describes the implementation of HRI using developed Lithuanian language synthesis and recognition tools. Lithuanian language has a very small population of native speakers, which makes training the synthesizers and automated speech recognition tools very complex. Different Lithuanian language dialects and non-native speakers also experience severely deteriorated language recognition performance. There are still many issues concerning the lower level of the HRI field. The current automated Lithuanian speech recognition does not take into account the speaker's emotions. The issues of HRI are being considered. The task of developing HRI control system is discussed. HRI with the NAO V6 robot, a control system for verbal interaction, is discussed. The contribution of our research is related to the application of HRI to the pedagogical problem of teaching and motivating school children in the classroom.

Keywords: control systems, humanoid robotics, human-robot interaction, Lithuanian speech recognition, Lithuanian speech synthesis.

1. Introduction

Robots are used in various fields (Romero-Gozalez et al., 2020 and Alibegovic et al., 2020). The number of humanoid robots, used to communicate with humans (Hedayati et al., 2023 and Kumazaki et al., 2023) and to solve problems in healthcare (Yoshida et al., 2022), entertainment industry and education is increasing (Brinckhaus et al., 2021). Humanoid robots are being used to teach languages, practical engineering (Yi et al., 2016), nutrition (Rosi et al., 2016), mathematics (Baxter et al., 2017) and general science, and help students learn spelling, storytelling and participate in memory games. Educational humanoid robots (Gupta et al., 2021) have been used in various educational settings and have appealed to a wide range of students, such as preschool children, elementary school students (Chu et al., 2019), high school students and engineering students (Alnajjar et al., 2019). Children generally respond positively to the robots. A positive impact on learning (Baxter et al., 2017) has been observed as well as higher participation, an increase in a student's creativity, curiosity, knowledge and recall (Li et al., 2016). HRI would benefit greatly from the use of sentiment analysis, which can categorize students' emotions according to their type (positive, negative, neutral) and sometimes event intensity (very positive / negative, somewhat positive / negative, etc.) (Kapočiūtė-Dzikienė et al., 2022).

There is a growing need for systems that help humans and humanoid robots with artificial intelligence to communicate productively (Rosenberg-Kima et al., 2019). Emotion recognition is also being implemented in robots used for HRI. Emotion recognition based on human language has the most applications, not only in the context of direct communication between humans or human-computer interaction, but also in recognizing social cognitive abilities, diagnosing emotional deficits, etc. (Tamulevičius et al., 2020 and Tang et al., 2022). There are other problems with speech recognition because background noise is unavoidable in the real world. It significantly degrades the performance of speech-based applications, such as automatic speech recognition and speaker identification (Pandey et al. 2022).

In order to achieve productive communication, there should be a way to communicate in the native language. Nowadays, the use of information and communication technology has become important not only when the relevant content emerges, but also when there is a possibility of using computers and other devices in the mother tongue, especially between older people and children. In this paper, communication between humans and humanoid robots is solved with the help of verbal measures. The developed control system uses the humanoid robot NAO V6 (Glas et al., 2016). The theory, algorithms, hardware and software of the control system are presented.

As an example, the use of a verbal and non-verbal human-robot communication control system, using a humanoid with a communication control system to solve the problems of 7 – 10 year old children with skill development is presented. The humanoid robot communicating with the child educates the child, develops the child's skills by teaching the child what decisions the child must make when moving an object *Q* that is in state *A* through environment *S* to state *B* with a minimum of effort, time and material costs, resources. The child's education is achieved by communicating with the humanoid robot in Lithuanian speech recognition and synthesis engine (Laurinčiukaitė et al., 2018).

Our experience has shown that children have a great desire to communicate with humanoid robots. They communicate enthusiastically with the robot. Children are more interested in humanoid robots than in computers or virtual assistants. Communication in Lithuanian language is more natural, even though most children start learning English or other languages at an early age. Children who communicate with the robots, or even less motivated students, can start learning actively because of the desire to talk to humanoid robots. Using its programmed intelligence, the robot is able to engage children in the learning process, encourage them to actively participate, give them tasks that promote decision-making and provide the child with immediate feedback. By communicating politely, the humanoid robot also teaches children polite communication. Without the use of the imperative, which often occurs when giving instructions to other machines when the feedback is a completed task, a humanoid robot can form comments about correct and wrong answers by giving appropriate feedback.

2. Statement of the problem

Humanoid robots can help solve humanity’s problems. It would be convenient to communicate verbally with robots. There is a need to research the possibilities of HRI systems with verbal language. In this paper, the theoretical and practical aspects of HRI communication are analyzed when verbal communication is used. Communication tools based on the Lithuanian language tools are developed.

The HRI system consists of three components: humans, robots and communication tools that ensure HRI. The first component is the human being. It is not possible to precisely describe the characteristics of a human being. The characteristics of a human depend on their knowledge, their ability to use this knowledge, their experience and many other factors that cannot be fully controlled by other humans. For these reasons, we use probabilistic methods to describe the characteristics of a person.

The second component is the humanoid robot. The humanoid must have some intelligence – information that mimics the natural knowledge of the human who created or assembled the robot. The robot’s capabilities can also be described by the depth of its memory, which helps to utilize the history of decisions made by a person and helps the robot to give advice to the person. The humanoid robot must be able to recognize the questions it receives from a person. The robot must be able to transmit information to the communicating person via verbal speech signals. The third component is the communication tool that ensures human-robot communication using speech and signals.

3. Solving the problem

A situation is analyzed in which humans and robots communicate verbally in Lithuanian. A situation is being analyzed in which the humanoid robot NAO V6 with Gentoo Linux operating system functions locally without using wireless or wired computers. The elements of the control system are installed on the humanoid robot’s computer. The block diagram of the interaction system is shown in Fig.1.

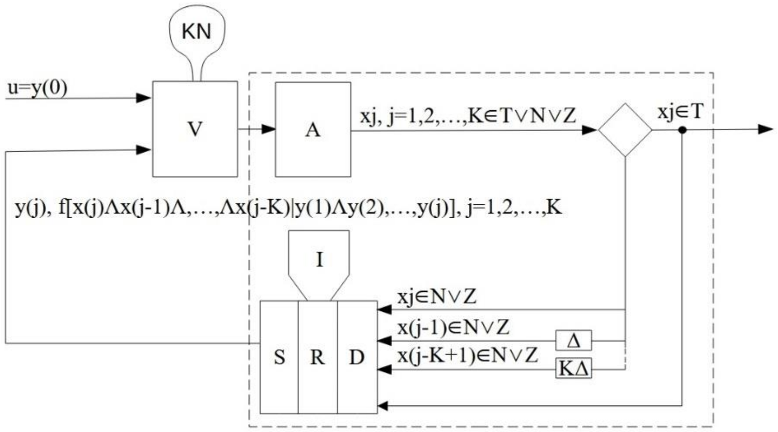

Figure 1. Block diagram of the developed verbal HRI using humanoid robot NAO V6

One of the elements of the control system is the human being – ***V*** with his intellect, with the knowledge ***KN*** in his head. A human is given a task ***u*** by a humanoid robot, which he must fulfil by interacting with the robot. The person can give the robot three types of answers: a correct answer, an incorrect answer or the sentence ***I do not know***. The second element of the control system is a humanoid robot ***R*** with intelligence ***I***, which gives the person advice on how to behave when solving the tasks set by the robot, which were created by the robot programmer. It is generally accepted that intelligence refers to the simulation of human intelligence in machines that are programmed to think like humans. The elements of verbal HRI using the Lithuanian language are: ***A*** – Lithuanian speech recognizer (Greibus et al., 2017), which recognizes speech and converts it into understandable commands, codes and symbols for the humanoid robot, ***S*** – Lithuanian speech synthesizer (Laurinčiukaitė et al., 2018), which gives suggestions or advice verbally in using Lithuanian. ***A*** and ***S*** are installed in the hardware located in the head of the humanoid robot. The third element of the control system is communication tool, which ensures the communication and interaction between humans and robots on a verbal level. The NAO V6 humanoid robot was chosen because it is cost-effective and enables flexible HRI.

Information about the decisions made by the person $x(t\text{-}k;\ j = N \vee Z)$ during the time $k = 0, 1, \dots K$. If the person answers ***I do not know*** $j = N$. If the person answers incorrectly – $j = Z$. If $x(t, t\text{-}1, \dots t – K; j{=}T)$ – means that the person has solved the given task ***u*** correctly.

If the person is unable to answer correctly more than three times, the humanoid robot thanks the person for the interaction, ends the interaction with this person and starts interacting with another person.

4. Human-robot communication system using NAO

This section describes an HRI verbal control system with the humanoid NAO V6 that can help a human develop their ability to make rational strategic decisions. It is important to note that the same principles of verbal communication can be used for other languages. However, when using other languages, the correct synthesizer and speech recognition engine should be used.

The computer of the humanoid robot NAO V6 consists of: ATOM E3845 – 1.91 GHz quad core GPU, 4 GB DDR3 RAM, 32 GB SSD. The robot has a 21.6 V 2.9 Ah battery, which is sufficient for an operating time of 1 hour. The robot has a network card that allows the robot to be connected via a Wi-Fi network or via an RJ45 Ethernet connection. The robot has 2 stereo speakers. The robot has 4 omnidirectional microphones that operate working in the range of 100 Hz – 10 kHz. The omnidirectional microphones are used to detect the direction of the sound source and receive the voice signals from humans. Inside the humanoid robot, 2 OV5640 cameras are integrated. The robot has LEDs around its two eyes and on both sides of its head. These LEDs are for information purposes and with their help you can observe when the robot is listening to the speaker, when it is fully functional and ready for work. The robot also has additional LEDs in its feet and chest. All LEDs can emit different colored lights – the eye colors can be green, red, yellow, blue, white or any other RGB color combination. The force-sensitive resistors are located in the robot's feet and their working range is between 0 and 25 N. The force-sensitive resistors are used to program the robot’s walking to determine when the foot is fully on the ground. The inertial module, which consists of a 3-axis gyroscope and a 3-axis accelerometer, is used to determine the current position of the robot, whether the robot is falling and at what speed it is moving. 2 ultrasonic sensors with a frequency of 40 kHz are used to measure distance, detect obstacles and communicate with other NAO robots standing in front of each other. The joints are equipped with 12-bit Hall effect sensors that are used to measure the rotation of the motors. The hands and the top of the head have capacitive sensors that act as touch switches. Both feet have bumpers, that function like buttons and are used as an additional measure to detect the obstacles in front of the robot instead of crashing into them. The humanoid robot has 25 degrees of freedom. The NAO V6 humanoid robot is 58 cm tall and weighs 5.5 kg.

Figure 2. Humanoid robot NAO V6 that was used in developing HRI in Lithuanian language

The humanoid robot NAO V6 was used to develop HRI applications for stimulating children’s creativity with elements of a communication control system. The system’s software was created using the visual programming language. Visual tools enable the programming of high-level robot behaviors by professionals and non-programmers. Visual programming languages have been used in industrial robotics for years. Their importance for social robotics has only recently been recognized when robots are to be used in therapy, education or other real-world applications (Baxter et al., 2017).

Although several works have dealt with the design principles for virtual programming languages in general, their application to social robotics has hardly been discussed so far. The virtual programming languages for social and home robotics that exist today differ greatly in their structure and feature set. Therefore, it is helpful to observe how the existing systems are used (Baxter et al., 2017).

A number of visual tools have been developed for robotics in general, although some are intended for tasks such as system configuration and are designed for programmers rather than non-programming end users (Baxter et al., 2017).

A widely used visual programming framework, especially for social robots, is Choregraphe, the Aldebaran software used to program the NAO and Pepper robots. Another visual language has been developed to enable non-programming therapists to program NAO robots, and RobotStudio is a visual programming environment designed to enable domain experts to develop applications for care robots, focusing on graphical interfaces on the robots touch panels (Baxter et al., 2017).

Choregraphe runs on Windows, Linux and MAC OS and offers some features for uploading content upload or monitoring. The implementation of NAO actions in Choregraphe is about connecting the elements of actions or movements (box) into a group that revolves around time or event.

A wide range of interactions are possible through wireless or wired communication, cameras, infrared sensors, microphones, loudspeakers and LEDs. The software structure is based on open source embedded Linux (Gentoo Linux) and supports programming languages such as C, C++, URBI, Python and .Net Framework. It also offers visual programming called Choregraphe.

Choregraphe is a cross-platform application that can implement NAO’s actions through visual programming. Unlike text-based programming, visual programming focuses less on grammar and programming is mostly done using the mouse rather than the keyboard to create nodes.

The programmed software robot decision control scheme is shown in Fig. 1. The human $\boldsymbol{V}$ receives a task $\boldsymbol{u}$ from a robot, that is a comment or an explanation followed by a question. The human must then tell the robot the answer, which is the decision. If the task is completed, the robot moves on to another task. If the task is not completed, the robot $\boldsymbol{R}$ gives the human a suggestion for a correct answer. After the human has listened to the suggestion, it must give an answer again.

The implementation was carried out with the Lithuanian speech synthesizer Liepa (Kasparaitis et al., 2023) and the recognizer LiepaASR for the humanoid robot NAO V6. Lithuanian is not one of the officially supported languages in the manufacturer’s list. Since the solution is based on artificial neural networks, which are widely used in speech signal synthesis today, it is essentially an aggregative approach – all knowledge about the generated speech signal is obtained by examining many speech examples, which can differ greatly in the contextual information they contain (Melnik-Leroy et al., 2022).

The functional diagram of the NAO V6 humanoid robot Lithuanian synthesizer is shown in Fig. 3. It is based on the unit selection method (Kasparaitis et al., 2023) and is programmed in the C++ language. Unit selection synthesis uses annotated recordings of speakers. The customized synthesizer for the robot has two voices: male and feminine. Each speaker recorded about 3 hours of speech. Next, the samples were divided into synthesis units (Kasparaitis et al., 2023) that synthesize the text. Based on the phonetic and prosodic characteristics of the text, the most suitable synthesis units or their sequences are found in the sound base, which are connected one after the other to turn the text into a sound. The text is normalized before synthesis begins, that is various abbreviations, numbers, dates and similar text are replaced by complete words. The Liepa synthesizer outputs audio signals at a sampling rate of 22kHz. Each language model requires about 500 MB of memory.

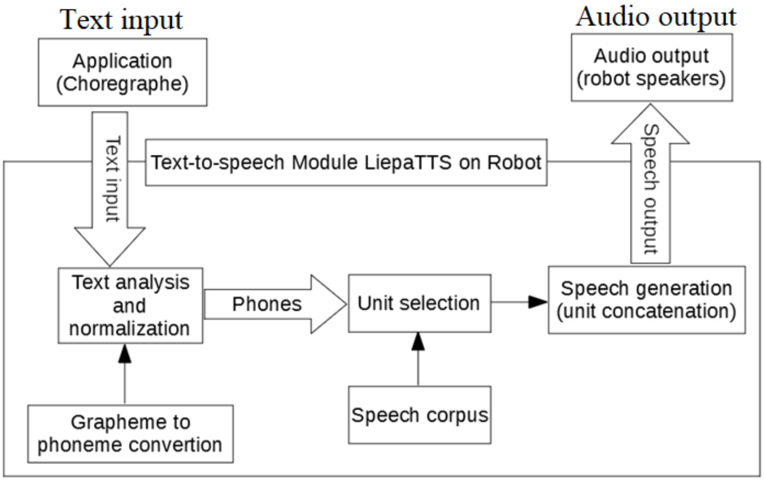

Figure 3. Speech generation engine system

The Liepa synthesizer installed in the NAO V6 robot has such functions:

- Synthesize text and play the sound through the speakers (text encoding must be UTF-8);
- Synthesize text and write the result as an audio file to the selected directory;
- It is possible to set or recall various parameter values that are currently used in the synthesizer: voice (male / female), voice speed, volume, pitch;
- It is possible to get a list of Lithuanian voices installed in the robot;
- Stop the current playback of the synthesized result;

The synthesizer also sends signals about events to the application, for example that the playback of the synthesized text has started, has ended, the word or sentence has started or has ended. These events can be used by other applications working in the robot.

It is possible to invoke the synthesizer and its functions from the Choregraphe environment, either from other robot applications or from a command line while connected to the robot.

The NAO robot system is equipped with a speech recognition module. It can process more than twenty different languages. This is very handy for creating learning scenarios, but we have found that the limiting factor is the extensibility of the language scope. The implementation was made for recognition modules for language families with a small market (Estonian, Latvian and Lithuanian).

The robot system offers the option of connecting audio inputs and outputs. The digital signal from the microphones can be processed internally by software modules. An additional speech recognition module was created based on an existing generic speech-to-text engine (Greibus et al., 2017) developed for the Lithuanian language. This engine, shown in Fig. 4, can be extended and used for many other languages without additional restrictions, such as: Latvian and Estonian.

The robot module was written using in the C++ language as an adapter for the Pocketsphinx speech recognition program, which uses an acoustic language model created as part of the LIEPA project. We have also added utilities that allow the user to create language models for educational scenarios and dictionaries (grapheme-to-phoneme) by users with no technical knowledge of automatic speech recognition. The module runs on the robot's operating system and does not require a network connection. It provides the ability to recognize commands required to perform a step in a particular scene.

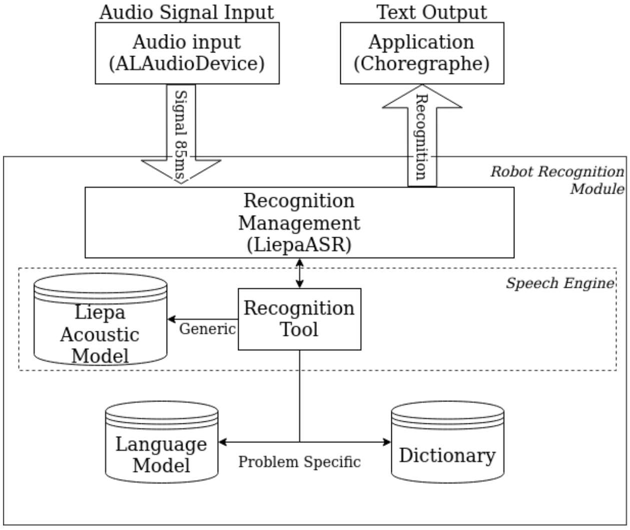

Figure 4. Lithuanian speech recognition engine system used in NAO V6 robot

The module works asynchronously, that is it works in parallel with other sensor and programming logic. The module subscribes to audio signals from the robot's event management system. It is predefined that the signals are 1365 samples long (16 kHz, 16 bit) and the acoustic model requires about 20 MB of memory. The speech module manages the status of the recognition tools and selects the correct speech model and dictionary when required. When the engine provides recognition results, a new event is generated by the application via the robot event management system. With this module there are infinite possibilities to create HRI scenarios (Greibus et al., 2017).

To program the NAO's behavior, we used Choregraphe, a visual development environment provided by the robot manufacturer Aldebaran Robotics. Choregraphe is one of the most important software tools used when working with the NAO robot. With Choregraphe it is possible to create programs, write dialogs or adjust the behavior of NAO. The interface is mainly drag and drop and allows the expert to create a sequenced combination of predefined or user-defined behavior boxes to manipulate the NAO's joints or attributes.

To enable the robot to interact with the students, applications were implemented with Choregraphe, which were installed in the internal memory of the NAO V6 robot. Custom behaviors were created that combine simple actions with a text-to-speech engine. The robot application itself was written on a computer using the official Aldebaran robotics tool Choregraphe. In this tool, the entire robot application is defined using robot activities that describe the robot's actions, start conditions and reaction to signals. The developed robot application is uploaded and installed in the robot via the Choregraphe tool. The operating system of the NAOqi robot has a module of the autonomous launchpad that starts and executes the robot application with the programmed activity or activities. The output of the robot depends on the programmed activity. The process described for uploading the application is shown in the Fig. 5.

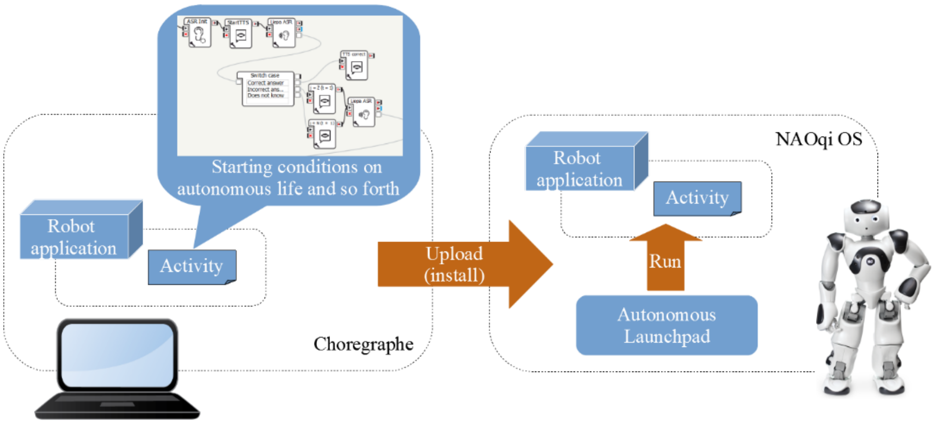

Figure 5. Running a humanoid robot NAO V6 application defined as an activity

The created robot software starts from the *Start* output, which is connected to the *LiepaASR*Init module. This module receives a signal at its *Start* input, which gives a signal to start the *LiepaASRInit* module. Once the robot has finished, the *LiepaASRInit* module stops and sends a signal from its *Stop* output to the *Start* input of the *Speak* module. The robot says the programmed text and then the module stops. As soon as the module has stopped, – it sends a signal from the *Stop* output to the *Start* input of the *LiepaASR* module. The *LiepaASR* module is responsible for recognizing words and phrases. As soon as words or phrases have been recognized, *LiepaASR* stops and sends the recognized text from the *onEvent* output to the input of the *Switch* module. The *Switch* module compares the received text with the list of possible responses. As soon as it has found the answer, it sends a signal via the output next to the recognized answer.

If the answer is correct – then the switching module activates the output next to the "correct" answer, which is connected to the speech module called "correct". If the correct answer was given, the robot says the programmed text and finishes its work. If the switch module has received incorrect answers, – they are connected to the corresponding modules, which in turn are connected to the *LiepaASR* module. The robot makes comments, says its programmed text and continues to listen to the human's answers. The program ends either by saying the correct answer or by saying an incorrect answer 3 times.

The official programming software of the robot manufacturer Choregraphe was chosen. Choregraphe is an official software development tool used for programming the behavior of the humanoid NAO using visual programming. In the software, everything is programmed via events that are linked to other events in various connections and dependencies.

The robot application that controls the behavior of the humanoid robot NAO is started by activating the *Start* output, which is located at the top left and is called *onStart*. This output only sends a signal to start the programmed application. The *OnStart* output sends a signal when it receives a command to run the software. It is started by double-clicking the *onStart* output or by uploading the software to the robot, and the autonomous launchpad in the robot automatically sends a command to start the robot behavior, as shown in Fig. 5. At the beginning of the software, no data other than the start command is sent.

The *LiepaASRInit* module is a module that has 2 inputs: *onStart* – the module starts when it receives a signal at this input and *onStop* – the module is stopped when it receives a signal at this input. It also has 1 output signal: *onStopped*, which is activated as soon as the module has stopped or completed its task. The *LiepaASRInit* module is responsible for the correct initialization of the *Liepa* speech recognition engine. The module describes the correct grammar and vocabulary paths required for the *Liepa* recognition engine to correctly recognize Lithuanian words and sentences. By default, the Choregraphe software does not support the Lithuanian language in its official software. For this reason, the initialization of the Lithuanian language engines and additional actions are required to ensure the correct recognition of the Lithuanian language in the robot.

The *LiepaASRInit* module also has Python code that is used to describe the module, the required parameters and the paths to the required files. The *LiepaASRInit* module is only needed for a correct start of the Lithuanian speech recognition engine. The module itself does not receive or send any data or parameters. As soon as this module receives a signal at its *onStart* input, it starts, sets the correct parameters and paths for the Lithuanian speech recognition engine and stops after completing the tasks and sends a signal via the *onStopped* output.

The speech module has 2 inputs: *onStart* and *onStop*. It also has 1 input signal *onStopped* which is activated when the module has completed its task. In the speech module, the text to be spoken by the robot is written in Lithuanian characters using the Python code. The module starts working when it receives a signal at its *onStart* input. As soon as the module has spoken the entire written text in Lithuanian, it stops and automatically sends a signal at the *onStopped* output.

LiepaASR is a module that has 2 inputs: *onStart* and *onStop*. It has 3 output signals: *onStopped*, which are activated when the module has stopped working; *onError* – this output is used when an error occurs in the module and sends the information about the error in the form of text; *onEvent* – an output that sends the recognized words or phrases in the form of text. The *LiepaASR* parameters are described using the Python code within the module.

The switch module is a module that has 1 input *onInput*. This module has 4 output signals: *onDefault*, if the recognized words or phrases are not described in the list and there are three additional outputs *output1*, *output2*, *output3*. This module receives text in the input and then compares it with the words or phrases in the list – the output next to it is activated. The possible options are written with the simple text between quotation marks. The switching module can also have several outputs, depending on the list of possible answers.

Several situations have been analyzed in which HRI has been used with 7 – 10 year old students to stimulate skills and creativity in making rational decisions taking into account the criteria of the task.

In general, any educational behavior could continue *K* cycles. In this case, the robot would have to give advice to the human, taking into account all the decisions made in the previous cycles ($k = 0, 1, \ldots K$).

During the experiments it was found that the automated lessons encouraged interest in learning and that, people enjoyed interacting with the humanoid robot. Students actively participate in the learning process, they learn to make rational decisions, find the correct answers and also learn to concentrate to hear the task and articulate the words correctly, which is a common problem nowadays.

Such an HRI system can be used not only for developing various skills of students, but also for shopping malls, museums, libraries and guest services in hotels.

5. The experiment

43 people took part in the experiment. 82 % of the participants were male and the majority of the participants were between 14 and 23 years old. There were also 2 female participants aged up to 45 years. All experiments had the exact same settings, as shown in Fig. 6.

The majority of the experiments were conducted in the lecture classroom of Vilnius University. The room is not designed to minimize outside interference. The room has a rectangular shape with dimensions of 8 meters by 15 meters and a ceiling height of about 4 meters. There was no soundproofing or additional measures to reduce the noise coming from outside. The walls of the room are made of concrete, the windows are double-glazed and closed to minimize outside noise. The floor in the room was covered with laminate and the rooms are furnished with wooden and steel furniture, such as tables and chairs. During the interactions, all ventilation systems and temperature control devices in the room were switched off. The room temperature was around 21 degrees Celsius. Due to the room settings, the robot occasionally heard echoes during the experiments.

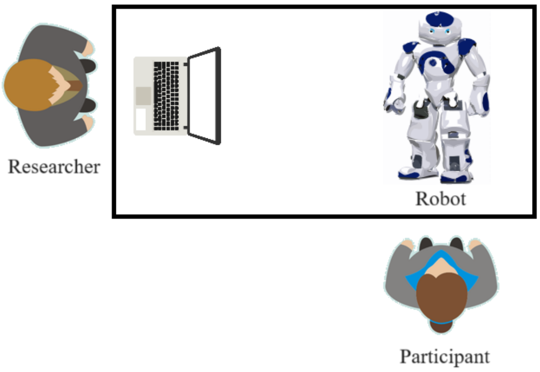

Figure 6. Experimental room setting

The humanoid robot NAO V6 was placed on a table in front of a human. The robot would use its internal microphones and speakers to interact with the human. During the HRI, the robot was active and would move its joints, creating additional external sounds. The human with whom the robot interacted sat at a distance of 50 centimetres to 1 metre in front of the robot. The robot's microphones were not calibrated before the experiments.

The experiments lasted a year, as it was difficult to find willing participants who wanted to take part in the experiments. Almost all participants had no previous experience with humanoid robots such as the NAO V6. Before the experiment, all participants were instructed by the researcher on how to properly speak and interact with the robot and what to do if communication failed. Only the participant would operate the robot and the researcher would not intervene.

The experiment was conducted in different locations with very different acoustic properties, such as: a library, a laboratory, small rooms, classrooms, corridors and other real-life locations.

The robot would interact with the participants by teaching them the traffic rules in Lithuanian. The experiment would last about 35 to 45 minutes, depending on how quickly the participants would respond to the robot's questions and how often they would give the correct answers. The robot would explain the theory, ask for confirmation and give answers.

89 % of the participants successfully completed all the tasks set by the robot. The other participants became too impatient because the robot did not hear their answers or the robot did not understand the participants' answers.

Throughout the experiment, the error rate for individual words was over 17 %, which made the interaction with the robot very uncomfortable at times and very demotivating for the participants, as none of them had advanced knowledge of how automatic speech recognition machines work. All 11 % of the participants who did not fully complete the experiment gave up due to the awkwardness of the interactions – either the robot could not hear them due to background noise or the participants could not speak in clear Lithuanian as they either spoke too fast or did not pronounce the words correctly.

Although, some of the participants stopped the experiments, all participants were very engaged and were always interested in further functions of the robot, even besides the interaction in Lithuanian.

The engagement and results of the lessons were not measured, as most of the participants aged 14 and older had already covered the topic of traffic rules at school and had not improved significantly.

The same experiments were repeated with children aged 7 to 10 years. However, all children in this age group could not quite finish the 35 to 45 minutes of instruction with the robot because they ran out of patience and started asking the robot general questions and completely ignoring the interaction with the robot. This could be an area for further investigation. However the robot should be limited to 15-20 minute lessons for younger participants as they do not have the capacity to listen to a full 45 minute lesson.

6. Conclusions

HRI with NAO V6 verbal communication control system theory, principles and design methods are presented. The application of the HRI verbal communication system to enhance human creativity is presented. As described in the paper, with the developed control system, the NAO V6 robot can successfully interact with humans using the Lithuanian language. HRI would greatly benefit from incorporating more information about the person, such as: recognizing human emotions, both from sound and video, as well as using non-verbal speech cues, such as certain movements.

Adding new functions to the HRI would significantly improve the quality of communication. Adapting to the specific microphones of the NAO robot and eliminating the noise of the fans in the robot's head would improve the quality of automatic speech recognition. It would be possible to use the 4 omnidirectional microphones to recognize the sound source, filter out the echoes and help the robot to maintain eye contact with the person while the interaction is ongoing. Furthermore, additional emotion recognition and synthesis would significantly increase the motivation of people interacting with the robot.

For a more pleasant HRI experience, it is necessary to develop local dialog tools in Lithuanian that can deal with incomplete sentences and unknown states in the spoken utterances and have the ability to complete or fill in the missing information when the speaker omits words.

References

Alibegovic B., Prljaca N., Kimmel M., Schultalbers M. (2020). Speech recognition system for a service robot – a performance evaluation, *The 16th International Conference on Control, Automation, Robotics and Vision ICARCV 2020*. DOI: 10.1109/ICARCV50220.2020.9305342.

Alnajjar F. S., Renawi A. M., Cappuccio M., Mubain O. (2019). A Low-Cost Autonomous Attention Assessment System for Robot Intervention with Autistic Children, *International Journal of Social Robotics*, 13. DOI: 10.1007/s12369-020-00639-8.

Baxter P., Ashurst E., Read R., Kennedy J., Belpaeme T. (2017). Robot education peers in a situated primary school study: Personalization promotes child learning, *PloS one*, 12(5).

Brinckhaus E., Barnech G. T., Etcheverry M., Andrade F. (2021). RoboCup@Home: Evaluation of voice recognition systems for domestic service robots and introducing Latino Dataset, *2021 Latin American Robotics Symposium (LARS), 2021 Brazillian Symposium on Robotics (SBR), and 2021 Workshop on Robotics in Education (WRE)*, 25-29. DOI: 10.1109/LARS/SBR/WRE54079.2021.9605485.

Christodoulou D. (2017). Making Good Progress? The Future of Assessment for Learning? *Oxford University Press-Children*.

Chu j., Zhao G., Li Y., Fu Z., Zhu W., Song L. (2019). Design and Implementation of Education Companion Robot for Primary Education, *2019 IEEE 5th International Conference on Computer and Communications (ICCC),* 1327-1331. DOI: 10.1109/ICCC47050.2019.9064253.

Gupta M., Jain A. (2021). Challenges of Robot Assisted Teaching in Education Domain, *2021 International Conference on Computational Intelligence and Computing Applications (ICCICA),* 1-4. DOI: 10.1109/ICCICA52458.2021.9697252.

Romero-Gonzalez C., Marinez-Gomez J., Garcia-Varea I. (2020). Spoken language understanding for social robotics, *Proceedings of 2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. DOI: 10.1109/ICARSC49921.2020.9096175.

Rosenberg-Kima R., Koren Y., Yachini M., Gordon G. (2019). Human-Robot-Collaboration (HRC): Social Robots as Teaching Assistants for Training Activities in Small Groups, *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 522-523. DOI: 10.1109/HRI.2019.8673103.

Tamulevičius G., Korvel G., Yayak A. B., Treigys P., Bernatavičienė J., Kostek B. (2020). A Study of Cross-Linguistic Speech Emotion Recognition Based on 2D Feature Spaces, *Electronics*, 9. 1725. DOI: 10.3390/electronics9101725.

Glas D. F., Kanda T., Ishiguro H. (2016). Human-robot interaction design using Interaction Composer eight years of lessons learned, *11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 303-310, IEEE.

Greibus M., Ringelienė Ž., Telksnys L. (2017). The phoneme set influence for Lithuanian speech commands recognition accuracy, *Open Conference of Electrical, Electronic and Information Sciences (eStream)*, Vilnius, 1-4, DOI: 10.1109/eStream.2017.7950321.

Hedayati H., Seo S. H., Kanda T. (2023). Symbiotic Society with Avatars (SSA): Beyond Space and Time, *In Companion of the 2023 ACM/IEEE International Conference on Human-Robot Interaction (HRI'23)*, 953-955. DOI: 10.1145/3568294.3579964.

Kapočiūtė-Dzikienė J., Salimbajevs A. (2022). Comparison of Deep Learning Approaches for Lithuanian Sentiment Analysis, *Baltic Journal of Modern Computing*, **10**(3), **283-294,** DOI: 10.22364/bjmc.2022.10.3.02.

Kasparaitis P., Antanavičius D. (2023). Investigation of Input Alphabets of End-to-End Lithuanian Text-to-Speech Synthesizer, *Baltic Journal of Modern Computing*, **11**(2), 285-296, DOI: 10.22364/bjmc.2023.11.2.05.

Kumazaki H., Muramatsu T., Yoshikawa Y., Matsumoto Y., Takata K., Ishiguro H., Mimura M. (2022). Android Robot Promotes Disclosure of Negative Narratives by Individuals With Autism Spectrum Disorders, *Frontiers in Psychiatry*, 13. 899664. DOI: 10.3389/fpsyt.2022.899664.

Laurinčiukaitė S., Telksnys L., Kasparaitis P., Kliukienė R., Paukštytė V. (2018). Lithuanian Speech Corpus Liepa for development of human-computer interfaces working in voice recognition and synthesis mode, *Informatica*, 29(3), 487-498.

Li J., Kizilcec R., Bailenson J., Ju W. (2016). Social robots and virtual agents as lecturers for video instruction, *Computers in Human Behavior*, 55, 1122-1230.

Melnik-Leroy G. A., Bernatavičienė J., Korvel G., Navickas G., Tamulevičius G., Treigys P. (2022). An Overview of Lithuanian Intonation: A Linguistic and Modelling Perspective, *Informatica*, 1-38. DOI: 10.15388/22-INFOR502.

Oishi Y., Kanda T. (2017). Toward end-user programming for robots in stores, *In Proceedings of the Computation of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, 233-234.

Pandey A., Wang D. (2022). Self-Attending RNN for Speech Enhancement to Improve Cross-Corpus Generalization, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30. 1-1. DOI: 10.1109/TASLP.2022.3161143.

Rosi A., Dall' Asta M., Brighenti F., Del Rio D., Volta E., Baroni I., Scazzina F. (2016). The use of new technologies for nutritional education in primary schools: a pilot study, *Public Health*, 140, 50-55.

Tang S., Luo Z., Nan G., Baba J., Yoshikawa Y., Ishiguro H. (2022). Fusion with Hierarchical Graphs for Multimodal Emotion Recognition, *2022 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 1288-1296. DOI: 10.23919/APSIPAASC55919.2022.9979932.

Yi H., Knabe C., Pesek T., Hong D. W. (2016). Experimental learning in the development of a DARwIn-HP humanoid educational robot, *Journal of Intelligent and Robotic Systems*, 81(1), 41-49.

Yoshida A., Kumazaki H., Muramatsu T., Yoshikawa Y., Ishiguro H., Mimura M. (2022). Intervention with a humanoid robot avatar for individuals with social anxiety disorders comorbid with autism spectrum disorders, *Asian Journal of Psychiatry*, 78. 103315. DOI: 10.1016/j.ajp.2022.103315.

Received September 1, 2023, revised February 20, 2024, accepted February 26, 2024