Automatic Speech Recognition for Human-Robot Interaction on The Humanoid Robot Barelang 7

Nurul Indah Mawaddah[1], Charlie Rolando Andrian Tamba[2], Donny Prasetya Hutagalung[3], Frans Danielius Nainggolan[4], Eka Mutia Lubis[5], Eko Rudiawan Jamzuri[6]

{nurulindah1504@gmail.com[1], tambacharliepro@gmail.com[2], donnyhutagalung7@gmail.com[3], mutia@polibatam.ac.id[4], danielius610@gmail.com[5], ekorudiawan@polibatam.ac.id[6]}

Department of Electrical Engineering, Politeknik Negeri Batam, Batam, Indonesia[1,2,3,4,5,6]

Abstract. This paper proposes a Human-Robot Interaction (HRI) system through voice commands. The study used the Barelang 7 humanoid dancing robot platform comprising 29 Degrees of Freedom (DoF). Automatic Speech Recognition (ASR) was developed using Python programming language and the VOSK Offline Speech Recognition toolkit to recognize five instructions spoken by random speakers. The voice recognition results in ASR are then sent to the robot via serial communication to realize movements according to spoken instructions. From tests conducted on five speakers randomly with 100 attempts, a Word Error Rate (WER) value of 0.096 was obtained. Meanwhile, testing the entire HRI system achieved a success rate of 92%. These results indicate that the HRI system has been successfully implemented on humanoid robots, although there are still some errors in the ASR system, that affect the overall performance of the HRI system. The results of this research contribute to the research and development of HRI systems, especially in humanoid robots, which are still not widely studied.

Keywords: Human-Robot Interaction, Automatic Speech Recognition, Humanoid Robot, HRI, ASR.

1 Introduction

Human-robot interaction (HRI) is a multi-disciplinary field that designs, understands, and evaluates robotic systems involving human and robot communication [1]. Currently, research in HRI is rapidly increasing because communication between humans and robots is essential. The HRI is used to develop collaborative robotic systems that can work with humans. In the HRI system, humans provide instructions to the robot through sound, visual signs, or movements. Then, those instructions must be properly executed by the robot. In advanced HRI systems, robots not only carry out instructions but are also required to provide action feedback to humans through the media mentioned above.

One of the studies related to HRI on humanoid robots has been conducted by [2], which develops emotional expression systems using LED lights and breathing simulators. This humanoid robot will visually express emotions to the user. The emotional expression helps the

ICAE 2023, November 07, Batam, Indonesia

DOI 10.4108/eai.7-11-2023.2342940

user understand the given message by the robot. Meanwhile, in the health sector, the HRI is also of great benefit. Implementing HRI on robots is helpful as a substitute for professional nurses in serving patients. An example of its implementation is the robot Exode [3], developed to interact with older people through natural conversation. The robot is designed to remind older people to take medications. In addition, this robot has a remote monitoring feature, which can be used to determine the condition of older people in real-time.

HRI through voice commands is quite widely developed for robots. Voice commands are considered more efficient and natural to implement. The HRI implementation based on voice is also starting to be used in industrial automation. For example, HRI in the manufacturing industry was developed by [4] called Voice-Controlled Production (VCP). VCP is an implementation of Automatic Speech Recognition (ASR) on Human-Machine Interface (HMI) devices used to control machines or robots in production. In this study, an ASR was designed to predict the speech of German words. This ASR was designed using a combination of a Time-Delay Neural Network (TDNN) and a Long Short-Term Memory (LSTM) network, which was trained to detect 33 commands at a frequency of 16 KHz. By implementing this VCP, the production efficiency increased by 67%.

In robotic systems, HRI with voice commands is generally developed to control the robot's movement. Research conducted by [5] involves developing wheeled robots that can be controlled via voice. In [5], ASR is developed for Android applications using the Google Voice Assistant service. Then, the output data from ASR is sent to the robot controller via Wi-Fi. This study used eight commands to control the robot's movement: moving forward, backward, turning left, turning right, opening the gripper, closing the gripper, raising the arm, and lowering the arm. The proposed ASR showed an accuracy rate of 95% from tests conducted by three speakers. Meanwhile, [6] developed ASR on Android through Android Inter-Process Communication Android Interface Definition Language (IPC AIDL). In this study, five voice commands were used to control wheeled robots. The command is spoken in three languages, namely Korean, English, and Vietnamese. This study combined online and offline ASR for the multi-language ASR. Offline ASR is developed using deep learning models trained using TensorFlow and Keras libraries. The deep learning model is compressed and optimized for the RockChip RK3399 Pro processor. The deep learning model is specifically designed to detect English voices. As for Korean and Vietnamese voices, online ASR uses Google Cloud AI services. From the results of tests conducted using 1850 recorded voices, the accuracy rate obtained was 95% in English and Vietnamese. As for detecting Korean representatives, the ASR accuracy rate is only 87%.

Meanwhile, the development of HRI on more advanced robots is carried out by [7] and [8]. Robotic systems designed by [7] are used to help people who are in isolation when contracting infectious diseases. In this study, ASR was combined with word separation to extract intent from the instructions given by the speaker. Audio signals are processed by Mel Frequency Cepstral Coefficients (MFCCs) feature extractors and further classified by Convolutional Neural Networks (CNN). In this study, the words tested to be expected were the words "robot", "bring", "carry", "stop", "cup", "paper", "towel", and "medicine". The proposed speech recognition has an accuracy rate of 96.9%. Meanwhile, the proposed method achieved a success rate of 88.75% for the overall level test on the robot.

On the other hand, research conducted by [8] develops HRI on servant robots. In this system, the sound signal is first preprocessed to eliminate noise and detect the presence or absence of the speaker's voice. This preprocessing stage uses the Gaussian Mixture Model (GMM) implemented on WebRTC. The preprocessed sound signal is filtered to separate the registered speaker's voice from other random speakers. This voice separation combines CNN, LSTM, and a Fully Connected Neural Network (FCNN). Then, the separated sound is predicted using a Convolution-Augmented Transformer. This system obtained a WER value of 5.3% from the tests conducted, with an accuracy rate of 89.3%.

In the studies discussed earlier, the implementation of ASR is generally only on wheeled robots driven by two rotary actuators. In addition, the commands given to the robot are pretty simple. They are just basic movement commands. On the other hand, there is still no implementation of ASR to support HRI in humanoid robots. The application of ASR to humanoid robots is quite challenging because humanoid robots are formed by complex joint actuators consisting of many degrees of freedom. Integrating ASR with the action of humanoid robot movement is also not easy because what is controlled is not only two actuators, such as wheeled robots, but many actuators.

This study will discuss the use of ASR in humanoid robots for HRI systems. The HRI can be applied to humanoid service robots to facilitate interaction with humans through conversation. As the first step of the research, in this study, the robot was designed to be able to move according to commands spoken by the speaker. The five voice commands instructing the robot are standing, sitting, waving, clapping, and dancing. In this study, ASR is proposed to use VOSK Offline Speech Recognition to predict speech in English. In addition, a system for forming robot movements with data recording is also proposed.

More details related to the discussion of this paper will be explained in the next section. Section 2 describes the robot platform used for testing, how to form movement in the robot, and the ASR system itself. Then, we continue with Section 3, which discusses how to evaluate the results of this study and discussion of the results obtained. The paper is closed with the conclusion and future work section in the Section 4.

2 Materials and methods

This section will describe the methods used to implement human-robot interaction. First, we will explain the humanoid robot platform used in this study. Then, explain the method of forming movement patterns in robots. After that, it continued with an explanation of the speech recognition system implemented in the robot as well as how to integrate the system as a whole. The details of the explanation will be described in the sub-section below.

2.1 Humanoid robot platform

This research used the Barelang 7 humanoid robot platform as research material. The visualization of the Barelang 7 robot is shown in **Fig. 1**. The Barelang 7 is a humanoid robot for demonstrating traditional Indonesian dances. In addition, this robot actively competed in the Kontes Robot Indonesia (KRI), especially the Kontes Robot Seni Tari Indonesia (KRSTI) division. **Fig. 1** (a) describes the mechanical design of the robot, and **Fig. 1** (b) demonstrates the Barelang 7 robot equipped with traditional dancer clothes.

(a) (b)

Fig. 1. Humanoid dancing robot Barelang 7, (a) mechanical design and (b) robot performance with traditional dancer clothing.

This robot has a total number of joints with 29 Degrees of Freedom (DoF). Compared to designs from similar studies [9], the robot design proposed in this study has a more complex joint configuration. With this complex joint configuration, the robot can move more flexibly. The configuration of the robot parts can be explained as follows. Each leg has six joint actuators to move the torso so the robot can move to another position. Meanwhile, six joint actuators can imitate human dance movements on each hand. In addition, the movement of the robot's torso is also controlled by two additional joint actuators. These two joint actuators move the torso in the y-axis and z-axis. While in the head, three driving joints move the head towards the x-axis, y-axis, and z-axis.

The hardware configuration used to implement HRI through speech recognition can be seen in **Fig. 2.** Generally, the OpenCR microcontroller board processes all actuators on the robot. On this OpenCR board, an external EEPROM chip is added to store robot movement data. On the other hand, an Intel Nuc mini computer is separate from the robot to process the speech recognition algorithm from the microphone. Communication between the computer and the robot uses the RS-232 serial communication line. The data sent to the robot is text data, which is the output from the ASR system.

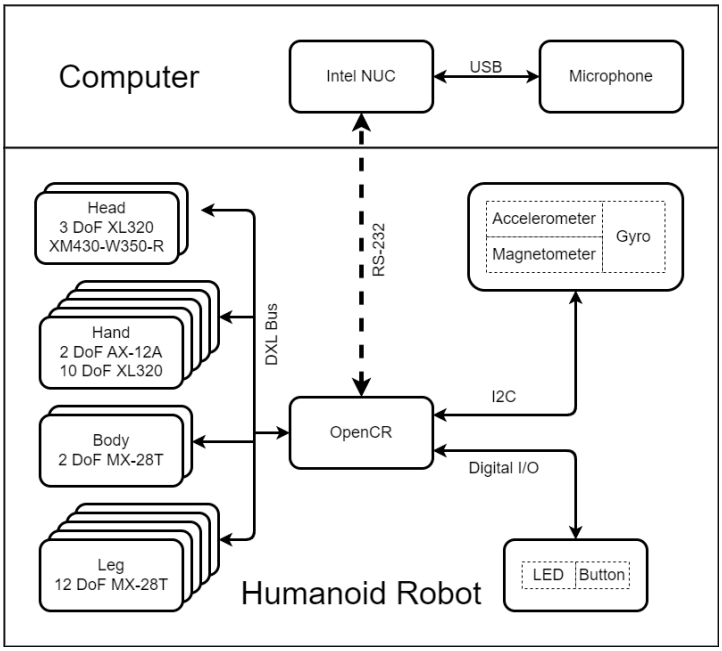

Fig. 2. Hardware block diagram of HRI implementation.

2.2 Motion generation

In this study, we designed a humanoid robot to be able to form five basic movements. These movements are standing, sitting, waving hands, applauding, and dancing. These movements are generated by all the joints located on the robot. Robots will later demonstrate these movements following instructions spoken by humans.

There are many methods for the formation of motion in humanoid robots. For stable walking movements, the preview control method is one technique that is often implemented. Such research was conducted by [10]; the walking pattern on the robot is formed by modeling the robot as an extended cart table. With this modeling, the Zero Moment Point (ZMP) trajectory on the robot is formed based on the footrest point that the robot will pass, or can be simplified by the footrest of the table. Furthermore, the ZMP trajectory forms a Center of Mass (CoM) trajectory using preview control techniques. After that, the inverse kinematics equation is used from the two trajectories to find the joint configuration in the foot.

In our research, the robot motion to be formed is static. The robot only performs motions on the spot without walking to another position. To form movements like this, we use the prerecorded motion method. The principle of this method is to store the values of the robot's joint configuration at a specific time in an Electrically Erasable Programmable Read-only Memory (EEPROM). Then, these values are used as a reference to form a trajectory for the entire joint of the robot. Unlike the motion formation technique proposed by [10], in this technique, the resulting trajectory is a trajectory in the joint space. The advantage of our proposed method is that there is no need for inverse kinematics equations to convert the world space trajectory to the joint space trajectory.

This joint space trajectory is formed based on the reference to each joint's starting position and ending position to be moved. Then, the trajectory is formed by mathematical functions such as polynomial curve equations [11] or piecewise quintic polynomials [12]. This mathematical equation aims to reduce jerks in the joints when moving. We use equation (1) to form the joint trajectory. In equation (1), $x(t)$ is the produced joint positions at the time-t. While x_i is the initial position of each joint and x_f is the final position of each joint. While d represents the total time spent to move the entire joint, and t is the current time.

$$x(t) = x_i + (x_f - x_i)(10(t/d)^3 - 15(t/d)^4 + 6(t/d)^5) \quad (1)$$

The joints controlled to form movements consist of 12 joints. These joints we note as $q_0 - q_{11}$. While $s_0 - s_n$ are the steps used to form the movement, and n is the number of steps stored in memory. This step has a different number for each movement, depending on how complex the robot moves.

The recording data for the joint angle in each movement is represented by the data in **Table 1** – **Table 5**. Joint angle recording data for sitting and standing movements are described in **Table 1** and **Table 2**. These two movements only require two steps, namely the initial and final positions when sitting and standing. As for the applause and waving motion, each step needed is three or four. Joint angle recording data for clapping movements is presented in **Table 3**, and joint angle recording data for waving movements is described in **Table 4**. Meanwhile, most steps needed for robot motion lie in the dance movement in **Table 5**. The robot needs 12 movement steps stored on the EEPROM for dance motion.

Table 1. Joint angle recording data for sitting motion.

Step number $(s_0 - s_n)$	Joint configuration (deg) $(q_0 - q_{11})$
s_0	(16.36, -67.04, 86.39, -39.24, -32.91 -17.07, 71.26, -91.67, 32.55, 37.47, -3.35, -11.79)
s_1	(6.86, 1.23, 2.99, -10.03, -7.57, -7.57, 1.58, -18.48, 9.67, 11.79, -2.64, -13.2)

Table 2. Joint angle recording data for standing movement.

Step number $(s_0 - s_n)$	Joint configuration (deg) $(q_0 - q_{11})$
s_0	(16.36, -67.04, 86.39, -39.24, -32.91, -17.07, 71.26, -91.67, 32.55, 37.47, -3.35, -11.79)
s_1	(6.86, 1.23, 2.99, -10.03, -7.57, -7.57, 1.58, -18.48, 9.67, 11.79, -2.64, -13.2)

Table 3. Joint angle recording data for clapping gestures.

Step number $(s_0 - s_n)$	Joint configuration (deg) $(q_0 - q_{11})$
s_0	(10.38, -38.89, 80.41, 23.4, -12.85, -7.57, 42.4, -80.76, -17.42, 23.75, -2.29, -7.92)
s_1	(10.03, -61.76, 92.02, 23.75, -12.5, -6.51, 74.42, -94.49, -18.48, 23.75, -2.99, -7.92)
s_2	(6.86, 1.23, 2.99, -10.03, -7.57, -7.57, 1.58, -18.48, 9.67, 11.79, -2.64, -13.2)

Table 4. Joint angle recording data for hand waving motion.

Step number $(s_0 - s_n)$	Joint configuration (deg) $(q_0 - q_{11})$
s_0	(16.36, -67.04, 86.39, -39.24, -32.91, -17.07, 71.26, -91.67, 32.55, 37.47, -3.35, -11.79)
s_1	(55.42, 37.47, 109.26, -16.37, -22.35, -1.23, -9.68, -7.92, 15.66, 13.19, 17.42, -13.9)
s_2	(70.9, 15.3, 131.78, -15.66, -22, -1.94, -10.03, -7.57, 15.66, 13.19, -9.68, -13.9)
s_3	(6.86, 1.23, 2.99, -10.03, -7.57, -7.57, 1.58, -18.48, 9.67, 11.79, -2.64, -13.2)

Table 5. Joint angle recording data for dance moves.

Step number $(s_0 - s_n)$	Joint configuration (deg) $(q_0 - q_{11})$
s_0	(18.12, -63.87, 76.54, -62.11, 42.4, -16.72, 67.39, -76.89, 50.49, -42.41, -4.75, -8.98)
s_1	(21.64, -58.59, 40.64, -58.95, -34.67, -30.44, 73.02, -36.07, 48.73, 32.55, -5.11, -7.57)
s_2	(16.01, -35.72, 79.35, 94.83, 43.46, -20.24, 39.94, -79, -112.79, -41.7, -5.11, -8.27)
s_3	(8.27, 94.48, 39.23, -112.44, -33.96, -38.54, -84.99, -57.19, 126.5, 18.47, -5.11, -4.75)
s_4	(46.98, 89.91, 14.6, -118.42, -39.94, -33.96, -91.32, -35.72, 99.76, 32.9, -2.99, -8.62)
s_5	(36.77, -24.81, 91.67, -47.69, 53.66, -36.07, 34.66, -91.67, 13.19, -45.57, -2.29, -1.94)
s_6	(17.77, -25.52, 99.06, -136.01, -36.42, -25.16, 31.84, -91.32, 126.5, 35.71, -23.76, -0.88)
s_7	(19.88, -20.59, 99.76, -136.01, -36.07, -22, 31.49, -91.32, 125.8, 35.71, 20.58, -1.23)
s_8	(33.25, -32.91, 99.06, -131.09, -43.81, -29.39, 20.93, -108.21, 149.03, 37.12, -5.81, -5.11)
s_9	(74.78, 74.07, 39.94, -114.9, -50.85, -29.39, 20.93, -108.21, 149.03, 37.12, -5.81, -4.75)
s_{10}	(39.23, -14.96, 112.43, -124.05, -31.15, -70.56, -88.51, -54.72, 126.5, 34.31, -7.57, -13.2)
s_{11}	(6.86, 1.23, 2.99, -10.03, -7.57, -7.57, 1.58, -18.48, 9.67, 11.79, -2.64, -13.2)

In terms of memory usage, the memory capacity needed to form one movement can be calculated using equation (2). Where M is the memory capacity needed, n_{steps} is the number of steps needed, and n_{joints} is the number of joints the robot is involved in when moving. The constant 8 Bytes is obtained from the total memory capacity needed to store data with the double precision type because each joint will store data in double.

$$M = n_{step} \times n_{joints} \times 8\ Bytes \tag{2}$$

From equation (2) above, the dance movement is the motion that requires the most memory storage capacity, which is 1152 Bytes. While the clapping and waving motions each require a memory capacity of 288 Bytes and 384 Bytes. Meanwhile, the movement that requires the least memory capacity is the sitting and standing movement, which is 192 bytes. So, it can be concluded that EEPROM is needed to store the entire movement with a minimum capacity of 2016 Bytes.

Fig. 3 illustrates the robot's movement when performing dancing, which has been stored as joint configuration data according to the data in **Table 5**. From **Fig. 3**, it can be seen that the robot will demonstrate different movements from each step of the joint angle configuration. As seen from s_0 until s_3, the robot moves to raise its hands and then tilts its body to the left. Then proceed with the step movement s_4 until s_7 by repeatedly swinging the body and hands in opposite directions. While the movement continued in steps s_8 until s_{11}, the robot began to move its head to add variety to the dance moves.

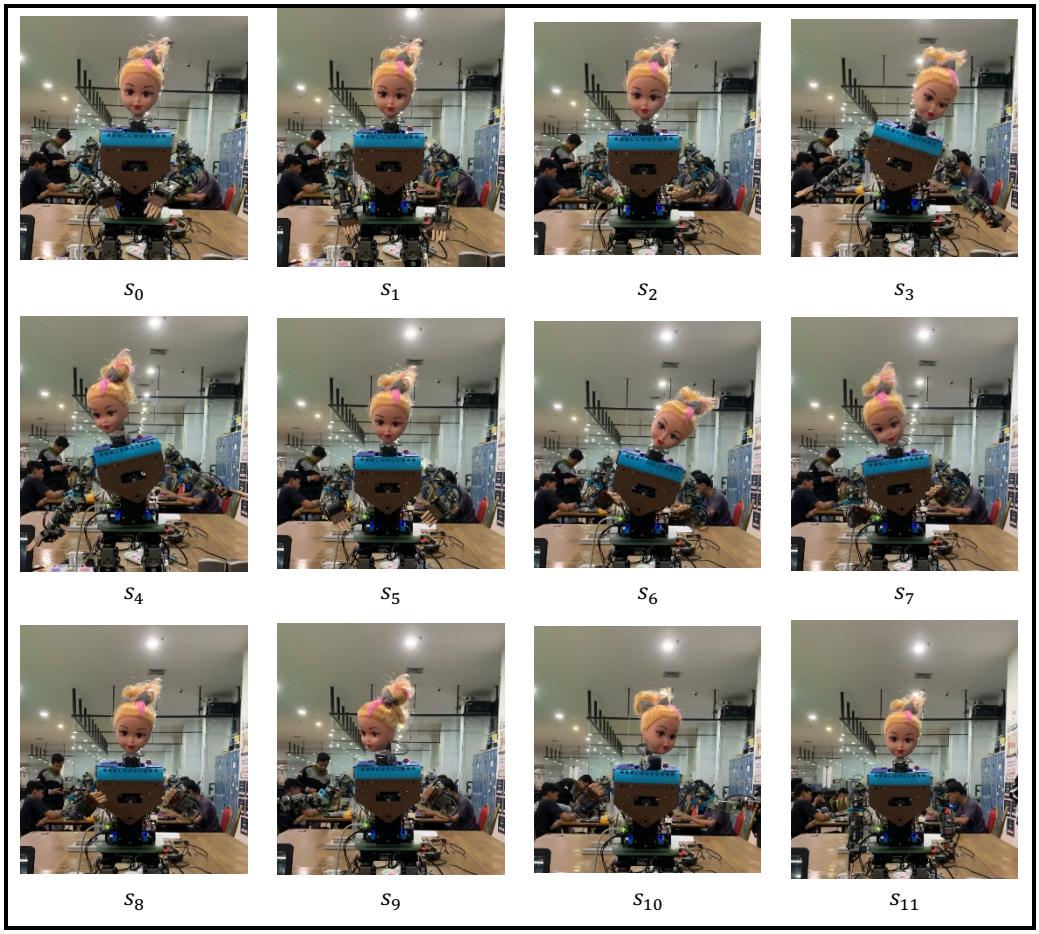

Fig. 3. Illustration of dancing motion on the humanoid robot Barelang 7.

2.2 Speech recognition

Automatic Speech Recognition (ASR) is a technology that translates spoken language from humans into text that computers can understand in real-time [13]. One application of ASR in everyday life can be found in smart homes. The latest smart home developed by [14] allows control of all house equipment using voice commands with the addition of ASR.

In general, two categories of ASR are widely implemented in research and development. The ASR category can be divided into online systems and offline systems. An online system is an ASR that requires third-party services to perform speech recognition inference. Online ASR system services generally provide an Application Programming Interface (API) that users can use to send voice signals to the server and receive prediction text from the server. Some examples of these online ASR systems include Facebook Wit.ai, Microsoft Azure Speech, Google Cloud Speech-to-Text, Wav2Vec, and AWS Transcribe [13]. This paid service is widely used in various studies, for example, in [15] and [16], which use Google services.

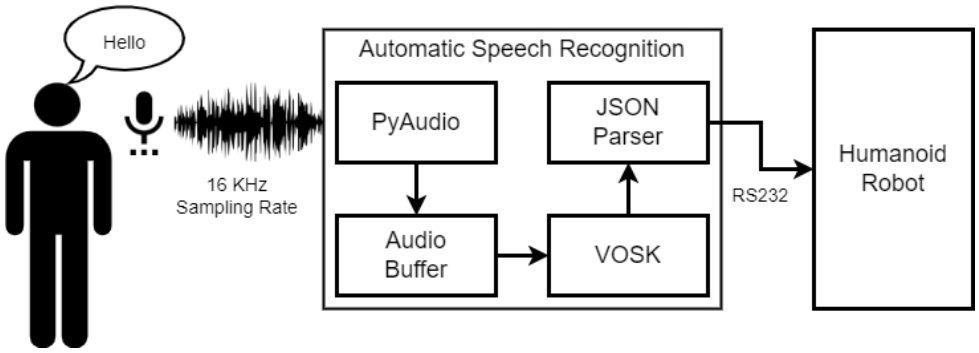

Fig. 4. Automatic Speech Recognition (ASR) on the humanoid robot Barelang 7.

Meanwhile, offline ASR systems do not require connections to third-party services. Offline ASRs are run directly on edge devices in real-time. For this offline ASR, popular deep learning-based algorithms can be implemented on edge devices with the help of several toolkits. Some popular ASR toolkits include the Hidden Markov Model Toolkit (HMM Toolkit), CMU Sphinx, Kaldi, DeepSpeech, VOSK, and LinTO [17], [18].

In this study, we used VOSK as the primary toolkit for the speech recognition process. VOSK is implemented using the Python programming language with an English speech recognition model on KaldiRecognizer. An overview of the speech recognition process is shown in **Fig. 4**. Meanwhile, to realize a real-time speech recognition system, we use the PyAudio library to acquire sound signals from microphones. Sound signal acquisition is performed at a sampling rate of 16 KHz. PyAudio will stream this audio signal and then store it in a buffer. This steamed audio signal is called a frame. A buffer can hold sound signals in more than one frame. In the case of this study, we used an audio buffer to hold 8192 frames of sound signals. We then took the first 4096 frames from that buffer to be detected using VOSK. The VOSK will predict spoken words from that voice frame, and then the ASR results in a JavaScript Object Notation (JSON) output format. The JSON data will then be parsed to take the text that contains predicted spoken words. This text is then transmitted to the humanoid robot through RS-232. Meanwhile, the data is translated into motion on the robot side by the method described in sub-section 2.2.

3 Results and discussion

This section will discuss the research results that have been obtained. The discussion will begin with an explanation of the evaluation methods in this study. In addition, we also explain how to conduct experiments to verify the results of this study. Then, we will describe the results obtained from several experiments. The explanations are described in sub-section 3.1 and sub-section 3.2 below.

3.1 Evaluation methods

We used two evaluation methods to test the proposed methods. Word Error Rate (WER) level measurement and Success Rate (SR) testing are the two test methods. WER testing was conducted to measure the error rate of voice recognition results. Meanwhile, the success rate is used to measure the overall robot performance in receiving and executing instructions given by the speaker.

The WER value can be calculated by comparing the words from the ASR prediction with the actual words spoken by the speaker [19]. Equation (3) is a formula for calculating WER. In equation (3), there is a notation S that describes the number of words exchanged, D which defines the number of words deleted, and I which states the addition of the word. Meanwhile, N is the actual number of words spoken by the speaker. The smaller WER value, the better the ASR system performance. Conversely, the higher WER, indicates the ASR is working poorly.

$$WER = \frac{S + D + I}{N} \tag{3}$$

Meanwhile, we tested the robot's success rate by following instructions using several random experiments. Then, calculate the robot's success rate in carrying out the instructions given using equation (4). We describe this level of success with the notation SR, where n_{ca} defines the correct actions executed by the robot according to the instructions. While n_c is the number of commands given to the robot during the testing. The higher SR value, the better the proposed system gets. On the contrary, the lower SR value describes the high failure rate.

$$SR = \frac{n_{ca}}{n_c} \times 100\% \tag{4}$$

3.2 Test results

Tests of the ASR are carried out by taking real-time sound signals from microphones and predicting them with VOSK. In this experiment, VOSK will print predictions of spoken words on the computer monitor. We then observe and compare the VOSK output with the actual words spoken by the speaker. The words used for testing are "sit down", "stand up", "hello", "dancing", and "clap". To improve the validity of the results, we tested the ASR system with five random speakers, consisting of three men and two women. Each speaker said the five words four times. So, the total number of experiments carried out was 100 times.

We gain the value WER=0.096 from the tests. In the test, the total word substitution was obtained (S), a total of 14; meanwhile, there was no addition of words (I) or word removal (D). The prediction of words containing errors is described in **Table 6**. In **Table 6**, the first woman to experiment is indicated by the symbol F1 and the second woman, F2; likewise, the first man, M1, the second man, M2, and the third man, M3. From the statistics, "dancing" is the word with the highest error rate. There were four prediction errors in this word. The phrase "dancing" is often predicted as the word "damn thing", "ben thing", or "then thing". In addition to "dancing", the word with a high error rate is "clap". The ASR often detects this "clap" as the words "club", "black", and "here is club". In addition, there is the word "stand up" spoken by M2, which is predicted as "spin up". However, this only happens once in the entire test.

Table 6. List of errors in ASR prediction results during testing.

Person	Spoken word	Recognized word
F1	dancing	damn thing
F1	dancing	ben thing
F2	dancing	damn thing
F2	clap	club
M1	clap	black
M2	dancing	then thing
M2	stand up	spin up
M3	clap	here is club

Table 7. Samples of spoken instruction and the robot action during testing HRI.

No.	Person	Spoken word	Recognized word	Robot action
1	F1	sit down	sit down	sitting down
2	F1	stand up	stand up	standing up
3	F1	hello	hello	waving hand
4	F1	dancing	dancing	dancing
5	F1	clap	clap	clapping
6	F2	sit down	sit down	sitting down
7	F2	stand up	stand up	standing up
8	F2	hello	hello	waving hand
9	F2	dancing	damn thing	no action
10	F2	clap	clap	clapping
11	M1	sit down	sit down	sitting down
12	M1	stand up	stand up	standing up
13	M1	hello	hello	waving hand
14	M1	dancing	dancing	dancing
15	M1	clap	clap	clapping
16	M2	sit down	sit down	sitting down
17	M2	stand up	stand up	standing up
18	M2	hello	hello	waving hand
19	M2	dancing	then thing	no action
20	M2	clap	clap	clapping
21	M3	sit down	sit down	sitting down
22	M3	stand up	stand up	standing up
23	M3	hello	hello	waving hand
24	M3	dancing	dancing	dancing
25	M3	clap	clap	clapping

After testing the ASR system, we tested the HRI system with the robot. In this test, we integrated the ASR system into the robot. So, we can observe the robot's movement and determine whether it matches the speech instructions. Supposedly, after hearing the "sit down", the robot moves to sit, and after hearing the "stand up", the robot moves to stand. Likewise, the robot's "hello", "dancing", and "clap" commands will move, waving, dancing, and clapping.

After testing 100 times, we found eight total failures, in which the robot did not perform any action after listening to the instructions. **Table 7** displays a selection of test results derived from 100 experiments. In this test, some failures are caused by ASR predictions that do not

match the list of motions in the robot. For example, in **Table 7** sample number 9, if speaker F2 says "dancing" but the ASR detects it as a "damn thing", then the robot will not perform any action because the "damn thing" motion is not stored on the EEPROM. With a total of 8 failures in this test, the n_{ca} value stands at 92. The number of n_c is equal to the number of tests, specifically 100 voice commands. So, from testing the system as a whole, an SR = 92% value was obtained as described in equation (5). Compared to similar tests conducted by [5], our success rate is 3% lower. However, regarding test validity, the number of speakers involved in this test was five, while in [5], there were only three. In addition, our proposed system is offline and does not require third-party services to perform ASR system inference. The results of testing the voice recognition system can be observed in the video demonstration at the following link: https://www.youtube.com/watch?v=Ab6OyGEY_gM.

$$SR = \frac{92}{100} \times 100\% = 92\% \tag{5}$$

4 Conclusion and future work

This study concluded that the HRI system using voice commands on humanoid dancing robots has been successfully implemented. This conclusion is evidenced by several tests carried out, both on ASR and the HRI system. The ASR proposed in this study has a WER performance of 0.096. This WER value shows that there is still potential for word recognition errors in the proposed system. This cause of error needs to be investigated further, whether it comes from mispronunciation by non-native speakers or due to deficiencies in the ASR. In the future, testing using native speakers must be done to conclude this. In terms of success rate, the robot can carry out movements according to instructions with a success rate of 92%. Some failures that arise are caused by the ASR, which is still inaccurate. Some parts still need to be explored, especially the method of forming robot motion. Storage of joint positions in EEPROM memory requires considerable memory capacity. It needs to be further investigated how this data is more efficient in storage so that the memory capacity required is efficient. Additionally, in the future, the proposed HRI system in the humanoid robot can be used to develop humanoid service robots that can interact directly with humans through conversation.

Acknowledgments.

This research is one of the outputs of the project-based learning and the student's final project in the Robotics Engineering Technology Study Program, Department of Electrical Engineering, Politeknik Negeri Batam. We thank the Politeknik Negeri Batam and Barelang Robotics and Artificial Intelligence Lab (BRAIL) for providing facilities and equipment resources to support research.

References

[1] R. R. Murphy, T. Nomura, A. Billard, and J. L. Burke, "Human–Robot Interaction," *IEEE Robot Autom Mag*, vol. 17, no. 2, pp. 85–89, Jun. 2010.

[2] Y. Xie and M. Matsumoto, "Emotional Expression for Humanoid Robot Using LED Light and Breathing Simulator," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 17, no. 9, pp. 1372–1374, Sep. 2022.

[3] J. M. Anson, L. Leo, M. T J, R. Milton, J. Davies, and D. Devassy, "Exode: Humanoid Healthcare Robot," in *2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, Mar. 2023, pp. 967–972.

[4] M. Norda, C. Engel, J. Rennies, J.-E. Appell, S. C. Lange, and A. Hahn, "Evaluating the Efficiency of Voice Control as Human Machine Interface in Production," *IEEE Transactions on Automation Science and Engineering*, pp. 1–12, 2023.

[5] M. Altayeb and A. Al-Ghraibah, "Voice controlled Camera Assisted Pick and Place Robot Using Raspberry Pi," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 10, no. 1, pp. 51–59, Feb. 2022.

[6] D. T. Tran, D. H. Truong, H. S. Le, and J.-H. Huh, "Mobile robot: automatic speech recognition application for automation and STEM education," *Soft comput*, vol. 27, no. 15, pp. 10789–10805, Aug. 2023.

[7] R. Jiménez-Moreno and R. A. Castillo, "Deep learning speech recognition for residential assistant robot," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 12, no. 2, p. 585, Jun. 2023.

[8] S.-A. Li, Y.-Y. Liu, Y.-C. Chen, H.-M. Feng, P.-K. Shen, and Y.-C. Wu, "Voice Interaction Recognition Design in Real-Life Scenario Mobile Robot Applications," *Applied Sciences*, vol. 13, no. 5, p. 3359, Mar. 2023.

[9] M. A. Fahd, D. Purwanto, and M. H. Fatoni, "Rancang Bangun Robot Penari Humanoid dengan Menggunakan 25 DoF untuk Melakukan Gerakan Tari Remo," *Jurnal Teknik ITS*, vol. 7, no. 2, pp. A362–A367, Feb. 2019.

[10] M. Szumowski, M. S. Żurawska, and T. Zielińska, "Preview Control applied for humanoid robot motion generation," *Archives of Control Sciences*, vol. 29, no. 1, pp. 111–132, Jul. 2023.

[11] G. Wu and S. Zhang, "Real-time jerk-minimization trajectory planning of robotic arm based on polynomial curve optimization," *Proc Inst Mech Eng C J Mech Eng Sci*, vol. 236, no. 21, pp. 10852–10864, Nov. 2022.

[12] S. Lu, B. Ding, and Y. Li, "Minimum-jerk trajectory planning pertaining to a translational 3-degree-of-freedom parallel manipulator through piecewise quintic polynomials interpolation," *Advances in Mechanical Engineering*, vol. 12, no. 3, p. 168781402091366, Mar. 2020.

[13] Z. Lin, "Self-correction of automatic speech recognition," *Applied and Computational Engineering*, vol. 5, no. 1, pp. 657–661, Jun. 2023.

[14] H. Isyanto, A. S. Arifin, and M. Suryanegara, "Design and Implementation of IoT-Based Smart Home Voice Commands for disabled people using Google Assistant," in *2020 International Conference on Smart Technology and Applications (ICoSTA)*, Feb. 2020, pp. 1–6.

[15] N. Anggraini, A. Kurniawan, L. K. Wardhani, and N. Hakiem, "Speech Recognition Application for the Speech Impaired using the Android-based Google Cloud Speech API," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 16, no. 6, p. 2733, Dec. 2018.

[16] H. J. Yoo, S. Seo, S. W. Im, and G. Y. Gim, "The Performance Evaluation of Continuous Speech Recognition Based on Korean Phonological Rules of Cloud-Based Speech Recognition Open API," *International Journal of Networked and Distributed Computing*, vol. 9, no. 1, pp. 10–18, Jan. 2021.

[17] A. Trabelsi, S. Warichet, Y. Aajaoun, and S. Soussilane, "Evaluation of the efficiency of state-of-the-art Speech Recognition engines," *Procedia Comput Sci*, vol. 207, pp. 2242–2252, Jan. 2022.

[18] A. Gupta, R. Kumar, and Y. Kumar, "An automatic speech recognition system in Indian and foreign languages: A state-of-the-art review analysis," *Intelligent Decision Technologies*, vol. 17, no. 2, pp. 505–526, Jan. 2023.

[19] T. von Neumann, C. Boeddeker, K. Kinoshita, M. Delcroix, and R. Haeb-Umbach, "On Word Error Rate Definitions and Their Efficient Computation for Multi-Speaker Speech Recognition Systems," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2023, pp. 1–5.