2023 IEEE International Conference on Robotics and Biomimetics (ROBIO) | 979-8-3503-2570-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ROBIO58561.2023.10354855

A VUI Foundation and Performance Validation for Voice-to-Motion Control of Snake-like Robots

Sean Casement[1], Yantao Shen[2] and Mengjun Zhang[1]

Abstract— **The locomotion of snakes is unique as it allows snakes to efficiently navigate complex and uneven terrains without articulated limbs. This movement pattern is attractive in the field of robotics exactly for its hyper redundancy, versatility, and lack of requirement for relatively complex locomotive systems. Snake-like robots that imitate these movement patterns are uniquely suited for use in extreme and hostile environments like deep sea exploration and outer space. Controlling robots in environments like these requires lots of training and intense concentration during operation. Voice User Interfaces (VUIs) can be used to bypass much of the need for this training and attention to operation by abstracting the direct control of a robotic system into the domain of speech. The design and implementation of a VUI demonstrating this idea is discussed here. The basic structure of a VUI is described and the implementation of a VUI in conjunction with a snake robot is shown. In experimentation the performance of the VUIs components were evaluated and the navigation of an obstacle course via the developed VUI and a remote controller were compared. The command recognition capability of the VUI was found to be 96% and the ability of the VUI to enable navigation of the obstacle course was found to be comparable considering the differences of the control formats.**

I. INTRODUCTION

Snake-like robots present a unique challenge in the domain of robotic control due to the many degrees of freedom that they possess. These degrees of freedom enable many modes of locomotion as well as many ways to interact with the environment as a manipulator. It is unreasonable to expect a human to be able to competently take advantage of this ability through a direct control interface, so it is especially relevant to the development of snake-like robots to implement systems that make their control less demanding on the skill of the user.

One way to reduce the complexity of control for a snake-like robot is the implementation of a Voice User Interface (VUI). VUIs are interfaces that allow users to interact with systems using only their voice. They combine several techniques like automatic speech recognition (ASR) and natural language processing (NLP) to convert spoken language into computer processes. Some common examples of VUIs are Amazon's Alexa, Google's Smart Home, Android Auto, Windows' Cortana, and even Microsoft Word's dictation tool.

*This work was partially supported by the NSF REU Site grant in Biomimetics and Soft Robotics with Award Number EEC #1852578.

[1]Sean Casement and Mengjun Zhang are with the Mechanical Engineering Department, University of Nevada, Reno, NV, 89557 USA scasement@nevada.unr.edu, mzhang@nevada.unr.edu

[2]Yantao Shen is with the Electrical and Biomedical Engineering & Mechanical Engineering Departments, University of Nevada, Reno, NV 89557 USA ytshen@unr.edu

Another common implementation of VUIs are voice controlled robots. VUIs are used in robotic control for their relatively low time investment to train new users. Training time is a common problem to be addressed when integrating a new control scheme into any workflow and delivering instruction by voice is a far more intuitive method of control that requires much less training to become proficient with than a remote controller. When used to direct robots a VUI's role is to translate human commands into robot intelligible instructions or goals. This can take many forms which range widely in complexity and scope depending on the capabilities and limitations of the systems. This can range from grammar-based systems that interpret the syntactic and semantic content of commands and extract command parameters that the robot can interpret to systems that can learn how to interpret commands using data sets that map between grounded actions a robot can take to language and other context clues in the environment [1][2][3].

Human-robot-interaction (HRI) is the field of NLP that is applicable to this project. It has a variety of uses from personal assistants like Amazon's Alexa and Google's Cortana to Microsoft Word's Dictate function. In the field of robotics, it is mostly used to facilitate the control of autonomous robots and provide a control scheme that is more intuitive to a user than a remote controller. When HRI is extended to robotic control it is used to translate natural language to robot understandable format like system parameters or instructions. One way to achieve that is by grounding tokens to the robot's systems and its environment directly. For example, when the word "move" is recognized in a sentence by the robot it can begin using NLP methods that translate the speech into valid parameters that can then be used by its locomotive systems. These direct connections enable NLP methods like regular expression analysis, syntactical analysis, and semantic analysis to be used to monitor incoming text for these tokens to trigger behavior or set goals depending on their presence.

In our previous work [4][5][6][7] the design and creation of the snake-like robot discussed in this paper were shown. Related to the work in this paper, Zinchenko *et al.* created a VUI for a minimally invasive surgical robot that utilized the ASR CMUsphinx to modify the duration of voice commands depending on the duration that the syllables in keyword commands were spoken for [8]. Contreras *et al.* also created a system that uses a similar technique to the one utilized here where the Levenshtein Distance is used in a VUI for a simulated UAV [9].

In our research we design and implement a VUI that grounds a dictionary of natural language commands to preconstructed actions that the snake-like robot or other robots can execute. Syntactic similarity as well as regular expression recognition are then used to parse between behavior and time duration clauses in commands recognized via wake-word recognition. To the best of our knowledge, this method of control for a snake-like robot has not been implemented previously though similar methods of control have been experimented with for other kinds of robots in simulation [9].

This paper is structured to describe the technical implementation of the VUI in Section 2. The decisions made concerning the techniques and tools used are described as well as the structure of the VUI. How the snake-like robot interprets the command parameters handed to it from the VUI is also described. Section 3 presents experimental setup and discusses the results found during experimentation. Section 4 offers our conclusions and some suggestions for future work on this project.

II. TECHNICAL METHODS AND IMPLEMENTATION

A. Operating Platform

Due to the computational requirements of the chosen ASR the VUI was developed and is implemented on a Raspberry Pi 4 B using the Aarch64 distribution of the Linux OS. The VUI was developed using Python for its rapid prototyping capabilities. A Raspberry Pi was chosen for its powerful computing power and support from the manufacturer and 3rd parties. The Linux OS was chosen due to its modularity and customizability for future optimization.

B. Vosk ASR

Several ASRs were considered for this project including Google Speech-to-Text, IFLYTEK short-form ASR, CMUsphinx, and Vosk. Google Speech-to-Text and IFLYTEK short-form ASR were discarded as options due to being closed source cloud-based ASRs. Investigation and experimentation with both ASRs found that the response time of cloud-based ASRs vary greatly depending on the quality of the internet connection to the servers. During experimentation with the IFLYTEK short-form ASR response times of the IFLYTEK servers were as low as 2 seconds and reported response times on the IFLYTEK website are as long as 1 minute [10]. Similar response times to our experimentation were reported by Google Speech-to-Text users in the Google forums ranging from 2 to 5 seconds. This much delay between the transmission and response from the cloud servers was not acceptable is why the cloud-based ASRs were eliminated from consideration. Vosk was chosen over CMUsphinx as the development team of CMUsphinx had discontinued development at the time to move onto more state-of-the-art ASR research while Vosk is still actively developed and supported by the creators [11][12].

The Vosk ASR was chosen for its preconstructed models designed for low power systems like single board computers and mobile applications and its open-source nature. The Vosk ASR is built on the Kaldi SDK which is an open source ASR development tool developed by a team at John Hopkins University [13]. In this project the Vosk ASR is implemented with the vosk-model-small-en-us-0.15 language model which is a preconstructed model intended for use on single board computers and in mobile applications.

C. Command Recognition via NLP

Regular expression analysis and syntactical similarity analysis are used in the VUI to scan incoming speech for wake-words and command clauses that are then parameterized into robot understandable commands. Regular expression analysis is used to search for root words that signify multiple command clauses. In the case of a relevant root word, the command clauses are separated and converted into command parameters. Syntactical similarity analysis is performed using the Normalized Generalized Levenshtein Distance (NGLD) defined in [14]. The Generalized Levenshtein Distance (GLD) is defined as the minimum edit cost required to transform string S into string T [15][16]. The GLD qualifies elementary edit operations as insertions, deletions, or the substitutions of a character for another in a string of characters where a character is any character in the alphabet Σ as well as the null string λ which is an empty string of length zero.

$$GLD_{s,t}(i,j) = \begin{cases} max(i,j) & if\ min(i,j) = 0 \\ min \begin{cases} GLD_{s,t}(i-1,j) + \gamma(\lambda \rightarrow a) \\ GLD_{s,t}(i,j-1) + \gamma(a \rightarrow \lambda) \\ GLD_{s,t}(i-1,j-1) + \gamma(a \rightarrow b) \end{cases} & if\ min(i,j) \neq 0 \end{cases} \quad (1)$$

Where i is equal to the length of string S and j is equal to the length of string T. Three edit operation costs are considered when calculating the Generalized Levenshtein Distance: insertions of characters denoted $\gamma(\lambda \rightarrow a)$, deletions of characters denoted $\gamma(a \rightarrow \lambda)$, and substitutions of characters denoted $\gamma(a \rightarrow b)$. These edit costs are minimized each time a new edit operation is performed. The edit costs used in the implementation are a cost of 1 for deletions and insertions and a cost of 2 for substitutions. In the case of a substitution where the ith and jth characters in strings S and T are equal the edit cost of substitution is 0. The Generalized Levenshtein Distance can be normalized using (2) provided (3) and (4) are held true.

$$NGLD_{s,t} = 1 - \frac{2 \cdot GLD_{s,t}}{\alpha \cdot (|S| + |T|) + GLD_{s,t}} \quad (2)$$

$$GLS_{s,t} = \frac{\alpha \cdot (|S| + |T|) - GLD_{s,t}}{2} \quad (3)$$

$$\forall a \in \Sigma, \gamma(\lambda \rightarrow a) = \gamma(a \rightarrow \lambda) = \alpha \quad (4)$$

The NGLD returns a value in the range of [0, 1] where a value of zero represents no syntactical similarity between the compared string and a value of 1 representing perfect syntactical similarity.

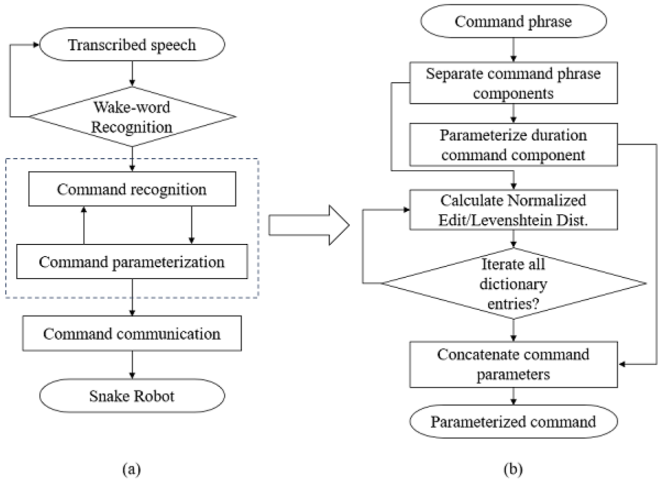

Fig. 1. Implemented Voice User Interface command recognition structure (a), command recognition and parameterization substructure (b)

D. VUI Structure

Figure 1 illustrates the structure of the VUI and how it handles the parameterization of new commands. The VUI uses wake-word recognition to pay attention for new commands from the ASR transcription. The wake word recognizer tokenizes the first two words of a transcription to check if they match a root wake-word. If a wake-word is recognized the transcription is handed to the VUI's command recognition methods.

Regular expression recognition is then used to separate the command phrase into two clauses. The VUI uses regular expression recognition to search for the root word "second" within the command phrase and if the root word is found then the command phrase is split into two strings with a rule-based grammar. The first string contains the behavior clause and the second string contains the time duration clause. If the root word "second" is not present in the command phrase then the VUI assumes that there is no time duration clause and moves on to the behavior clause recognition. In the case of a detected time duration clause the numeric word indicating the number of seconds the behavior should be executed for is converted to an integer and stored as a command parameter. Once the time duration command parameter has been stored the string containing the behavior clause is compared to a list of predefined commands using the NGLD. The list of predefined commands are grounded to parameters that indicate preprogrammed behavior on the snake-like robot. The predefined command that has the greatest similarity and is also over 75% similar to the behavior command clause is converted to the corresponding behavior command parameter. Both command parameters are then sent to the snake robot for interpretation and execution.

The VUI is connected to the snake-like robot via an XBee wireless radio frequency module through which the VUI sends command parameters to the snake-like robot. Upon receiving new command parameters, the snake-like robot ceases any behavior that it is current exhibiting and begins executing the behavior called for by the newly received command parameters. When a new command parameter is received by the snake-like robot it checks for a new behavior parameter and a new time duration parameter. The command parameter is parsed, and the corresponding behavior is executed. In the case of a time duration parameter, it is applied to the behavior being executed. If there is no time duration parameter, then the behavior is executed until new command parameters are received.

III. EXPERIMENTAL SETUP AND VALIDATIONS

Three tests are performed for this project. The first test is navigation of an obstacle course to reach a goal, the second determines the response times of the VUI and compares them to the remote controller, and the third examines the VUI controller's robustness to noise using Additive Gaussian White Noise (AGWN).

A. Obstacle Course Navigation

Experiment 1 tests the capability of the VUI controller to navigate the snake-like robot through an obstacle course to reach a goal. The course imitates a cluttered environment that could be found in a common operation area like a house. There are two mild turns and an about face leading to a straight path ending at the goal, a blue star, shown in Figure 2. Figure 3 and Figure 4 show the routes taken using the VUI and the remote controller from previous work respectively.

The route taken through the obstacle course begins at the position indicated by the yellow measuring tape seen in the top right of Figure 2 to the goal at the bottom. The course is divided into four sections marked by the blue lines. Table 1 details how much time the snake-like robot spends in each section using both control methods. The path that the snake-like robot follows was tracked using the Tracker image tracking software by tracing the first joint of the snake-like robot [17]. In Figure 3 each locomotion pattern change caused by the VUI is represented by a color shift in the curve. This clear demarcation of behavior is not indicated in Figure 4 due to the continuous nature of the input and configuration of the remote controller. The obstacle

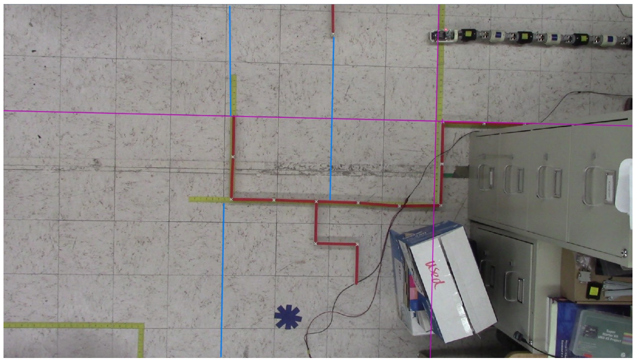

Fig. 2. Photo of the obstacle course with the snake-like robot place in starting position

course was completed in 25.833 seconds with the VUI and 70.004 seconds using the remote controller. The differences between the two paths taken are significant in two ways. The path taken over the first two sections is significantly more pronounced for the VUI. While the remote controller offers an easily variable turning radius with fine control it is limited in severity. the VUI, on the other hand, has been created such that as the snake-like robot turns the turning offset increases additively. These differences arise from the advantages and disadvantages of the respective control formats and enable the snake-like robot to make tighter but more pronounced turns when using the VUI.

TABLE I
TIME SPENT IN EACH SECTION OF THE OBSTACLE COURSE

	VUI	***Controller***
Left turn	5.833 sec (22.58%)	6.106 sec (8.72%)
Right turn	2.433 sec (9.42%)	5.506 sec (7.87%)
Left turn	15.267 sec (59.10%)	55.789 sec (79.69%)
Straight forward	2.300 sec (8.90%)	2.603 sec (3.72%)

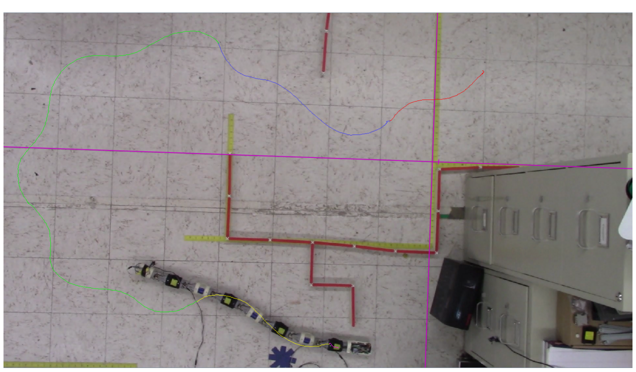

Fig. 3. Path of the snake robot through the obstacle course, controlled by the VUI system. Red: Turn left, Blue: Turn right, Green Turn left, Yellow: move forward, Magenta: Straighten up

The limited turning radius of the remote controller is the source of the other major difference between the two paths. It necessitated a multi-point turn which is the cause of the much longer navigation time as shown in Table 1.

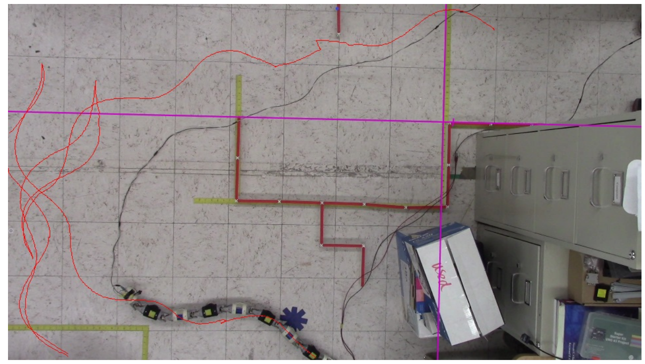

Fig. 4. Path of the snake robot through the obstacle course, controlled by the remote controller. The continuous path of the snake robot is represented by the blue line. Breaks in the line are due to the tracked feature on the snake leaving the camera frame.

B. VUI Response Time

In experiment 2 commands were issued to the VUI by the tester and the ASR transcription, command recognized, and NGLD between the transcription and recognized command are recorded. The response time of the snake robot is determined by timing the difference between the end of the spoken command and the beginning of the snake-like robot's reaction.

Figure 5 shows a whisker plot that details the distribution of the response times of commands with no time limit and commands with a time limit. Figure 6 shows the distribution of snake-like robot response times when using a remote controller.

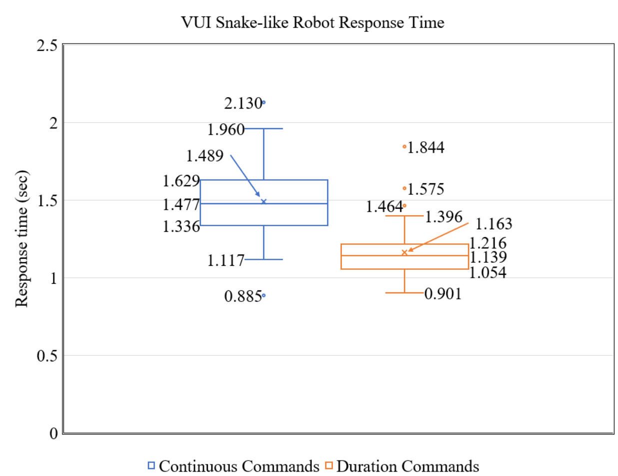

Fig. 5. Whisker Plot of the response times of the snake robot when controlled by the VUI system. The distributions are split between the commands with no time limit and commands with time limits.

The distribution of response times in Figure 5 varies significantly depending on type. The distribution of duration command response times is significantly different from the continuous commands and the average response time is 0.326 seconds shorter which is somewhat counter intuitive. The number of command parameters to process when executing a continuous command is less than the number

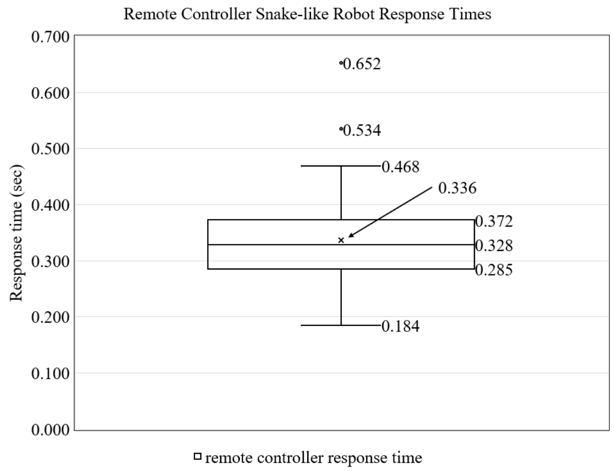

Fig. 6. Whisker plot of the response times of the snake-like robot when controlled by a remote controller.

for a duration command however, the construction of the snake-like robot control system causes the looping behavior function to check for new commands less frequently when continuously executing behavior. This is why the distribution of continuous command response times is larger than the duration commands and the continuous command response times are generally longer.

Figure 6 shows the snake-like robot response time distribution when operated with a remote controller. The distribution of response times is of much smaller scale in comparison to the VUI response time distributions and the average response time of 0.336 is much shorter as well. The duration commands fall short by 0.827 seconds on average and the continuous commands falls short by 1.153 seconds. This is a significant time difference and necessitates delivering a command via VUI further in advance than the remote controller to account for this response delay and the time required to deliver the command vocally. The smaller distribution of response times for the remote controller is also significant as it indicates the number of interpretive steps between remote controller input and behavior in comparison to the VUI distributions.

C. Reliability and Noise Robustness

Experiment 3 examines the Word Error Rate (WER) of the ASR, its ability to account for background noise, and the improvement over exact command recognition that the NGLD enables. AGWN is introduced to the audio input signal at several signal to noise ratios (SNR) and the WER of the transcriptions at these SNRs are compared. The SNR of a signal indicates the signal strength in comparison to the noise as shown in (5).

$$SNR = S - N \tag{5}$$

Where S is the signal strength and N is the strength of the noise, all in decibels. The SNRs used in this experiment are 40 decibels and 20 decibels respectively. The WERs of data sets at those SNRs are calculated and compared to the WER of a control data set. WER is the rate at which an ASR misstranscribes inputs as defined in (6). Where I is insertions, D is deletions, S is substitution, and N is the number of words in the data set.

$$WER = \frac{I + D + S}{N} \cdot 100\% \tag{6}$$

This experiment focuses on the command recognition ability of the VUI for the first command parameter which dictates behavior. This is due to the use of the NGLD to recognize commands for the first parameter and not the second. The second parameter uses regular expression analysis to detect when a time limit is specified as well as natural grammar constraints. During experimentation 288 data points were collected from the VUI. The distribution of the command recognition status of the data points are shown below in Figure 6.

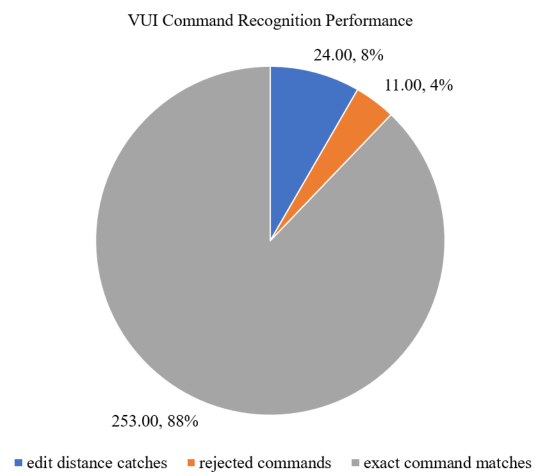

Fig. 7. The distribution of accepted commands based on an exact match with the dictionary of preconstructed commands and the NGLD.

Using the NGLD the rate of successful command recognition is 96%. A minimum NGLD of 0.75 was chosen for acceptance. 24 commands fell in between an NGLD of 1.00 and 0.75 and 11 commands fell below 0.75. The use of the NGLD increased the reliability of the behavior command recognition function by 8%. There are several factors that have a major effect on the WER of an ASR like the models and methods used by the decoder or external effects like pronunciation, positioning and number of microphones used, and background noise. The VUI implemented in this project uses the default offline Vosk ASR with the vosk-model-small-en-us-0.15 language model. The reported WER from the creators is 9.85%, found by testing with the Librispeech database [11]. Table 2 shows the WER found at different artificial SNRs using the commands from the list of known commands as well as the number of data points collected.

The control data is taken from the same data used in experiments 1 and 2 while the 20 and 40 decibel data were recorded in a quiet environment with minimal background noise. AGWN was added to the audio signal such that the

TABLE II

WER AT DIFFERENT SNRS

SNR	*Data points (words)*	*WER*
Control	1187	1.77%
20 dB	945	4.66%
40 dB	954	3.25%

desired noise level in the audio signal was achieved before it was handed to the ASR for transcription. The background noise for the control data originates from the lab that the experiment was performed in as well as from the snake robot during the experiment. The background noise of the control data set is the expected background noise to be present under normal operating conditions. The WER found from the testing data in this project is 1.77% which is significantly lower than the WER found via the Librispeech testing data listed on the Vosk ASR website. This can likely be attributed to the repetitive nature of the testing data and constrained vocabulary in the list of commands. There are several words and phrases that the ASR often mistranscribes. It tended to mistake the homophones "role" and "roll" as well as having difficulty transcribing words that undergo elision or reduction in speech like "straighten", "wiggly", and "winding". These kinds of problems are common to ASRs and are the main source of issues with the VUI and NGLD methods used. The WER of both SNR data sets are greater than the control data set which indicates that the SNR of the control data is greater than 40 decibels. This is a positive result as an SNR over 40 decibels is considered a relatively clear, noise free signal. A WER of 4.66% for a 20 decibel SNR is also a positive result as signals as that SNR level or below are considered relatively noisy. A possible source of error in this experiment is the nature of the noise present in the control data set in comparison to the others. The control data was collected in the lab under normal operating conditions where noise amplitude and frequency can vary freely and naturally with respect to the useful signal. This kind of noise is ordered in the sense that it originates from natural causes which can create a more continuous waveform. This ordered behavior is not how the AGWN in this experiment behaves; it is a generated randomly signal using a Gaussian distribution where the signal strength is manipulated to keep a constant SNR. This difference could influence how the feature extractor and acoustic model interpret the audio signal, changing the predicted senones.

IV. CONCLUSION

The development and validation of a VUI for use with a snake-like robot was demonstrated in this paper. The structure of the VUI, the techniques and tools used to develop it, and the performance of the VUI with respect to a remote controller where the advantages and disadvantages of both systems were discussed. The ability of the VUI to recognize commands that did not match the predesigned list of grounded commands was demonstrated as well as its robustness to noise.

The VUI developed for this project is a simple foundation that functions to replace the robots' control method with the feature that is more intuitively understandable. It offers an advantage in ease of use in exchange for greater response time to commands and less fine control over movement. In future projects the methods used in the VUI will be expanded to include semantic similarity analysis, rule-based clause segmentation, increased command parameter granularity as well as more commands, and intermediate task intuition. Efforts will also be made to address the robot response time delay in order to bring it more inline with the voice controlled robot responses.

REFERENCES

[1] E. Bastianelli, G. Castellucci, D. Croce, R. Basili, D. Nardi *et al.*, "Effective and robust natural language understanding for human-robot interaction." in *ECAI*, 2014, pp. 57–62.

[2] T. Kollar, S. Tellex, D. Roy, and N. Roy, "Toward understanding natural language directions," in *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2010, pp. 259–266.

[3] M. MacMahon, B. Stankiewicz, and B. Kuipers, "Walk the talk: Connecting language, knowledge, and action in route instructions," *Def*, vol. 2, no. 6, p. 4, 2006.

[4] G. Wang, W. Yang, Y. Shen, H. Shao, and C. Wang, "Adaptive path following of underactuated snake robot on unknown and varied frictions ground: theory and validations," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4273–4280, 2018.

[5] W. Yang, Y. Shen, and A. Bajenov, "Improving low-cost inertial-measurement-unit (imu)-based motion tracking accuracy for a biomorphic hyper-redundant snake robot. robotics and biometrics. university of nevada, reno, december 2017. 9 pp," 2019.

[6] W. Yang, G. Wang, and Y. Shen, "Perception-aware path finding and following of snake robot in unknown environment," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5925–5930.

[7] W. Yang, G. Wang, H. Shao, and Y. Shen, "Spline based curve path following of underactuated snake robots," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5352–5358.

[8] K. Zinchenko, C.-Y. Wu, and K.-T. Song, "A study on speech recognition control for a surgical robot," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 607–615, 2016.

[9] R. Contreras, A. Ayala, and F. Cruz, "Unmanned aerial vehicle control through domain-based automatic speech recognition," *Computers*, vol. 9, no. 3, p. 75, 2020.

[10] L. iFLYTEK Co., "Short form asr webapi document (automatic speech recognition)—iflytek open platform documents," https://global.xfyun.cn/doc/asr/voicedictation/API.htmldescription-of-the-interface, accesed:Oct. 23, 2021.

[11] A. Cephei, "Vosk offline speech recognition api," https://alphacephei.com/vosk/, accesed:Nov. 14, 2021.

[12] N. Shmyrev, "Cmusphinx open source speech recognition," https://cmusphinx.github.io/page4/, 2019, accesed:Nov. 11, 2021.

[13] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011.

[14] L. Yujian and L. Bo, "A normalized levenshtein distance metric," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 6, pp. 1091–1095, 2007.

[15] R. A. Wagner and M. J. Fischer, "The string-to-string correction problem," *Journal of the ACM (JACM)*, vol. 21, no. 1, pp. 168–173, 1974.

[16] V. I. Levenshtein *et al.*, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, vol. 10, no. 8. Soviet Union, 1966, pp. 707–710.

[17] W. C. D. Brown, R. Hanson, "Tracker video analysis and modeling tool," https://physlets.org/tracker/, accesed:May 6, 2022.