Voice Communication in Performing a Cooperative Task with a Robot

Koliya Pulasinghe[1], Keigo Watanabe[2], Kazuo Kiguchi[2], and Kiyotaka Izumi[2]

[1] Faculty of Engineering Systems and Technology,
[2] Department of Advanced Systems Control Engineering,
Graduate School of Science and Engineering, Saga University,
1-Honjomachi, Saga 840-8502, Japan.
†E-mail: koliya@ieee.org, {watanabe, kiguchi, izumi}@me.saga-u.ac.jp

Abstract. This paper investigates the credibility of voice (especially natural language commands) as a communication medium in sharing advanced sensory capacity and knowledge of the human with a robot to perform a cooperative task. Identification of the machine sensitive words in the unconstrained speech signal and interpretation of the imprecise natural language commands for the machine has been considered. The system constituents include a hidden Markov model (HMM) based continuous automatic speech recognizer (ASR) to identify the lexical content of the user's speech signal, a fuzzy neural network (FNN) to comprehend the natural language (NL) contained in identified lexical content, an artificial neural network (ANN) to activate the desired functional ability, and control modules to generate output signals to the actuators of the machine. The characteristic features have been tested experimentally by utilizing them to navigate a Khepera® in real time using the user's visual information transferred by speech signals.

1 Introduction

Use of robots as human assistants has been taken much attention in research community since they are very far from human dexterity to use them in place of human. In this context, robots are taken out from the manufacturing floor and used with humans in the human environment for the tasks like nursing and aiding where people can use them as assistants with limited functionality. A flexible communication medium is a must, where voice has the most plausible features among the others. Figure 1 describes the nature of the voice commands that we can use in performing a cooperative task. As described in the Fig. 1, the words used in natural conversations consist of lot of particles to maintain the grammatical structure of the uttered sentence and words having imprecise meaning. These two features can be emphasized as: 1) identifying the keywords which lead to activate the robot's functions (action words like lift) and 2) imprecise nature of words which describes how smoothly the robot should perform the particular action (fuzzy predicates like little, very slow). This paper proposes a methodology, which is capable

H. Asama et al. (eds.), *Distributed Autonomous Robotic Systems 5*

to represent above mensioned properties in voice based man-machine communication in performing a cooperative task.

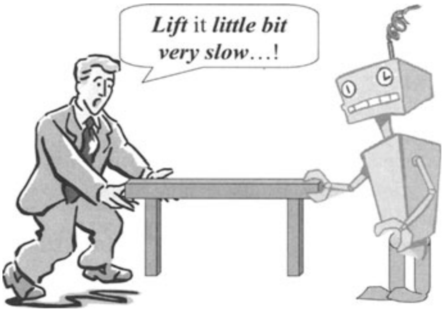

Fig. 1. Voice as the communication medium in a cooperative task

With the advent of fuzzy reasoning most researchers used it in the design of controller for a machine. But still fuzzy reasoning is not very popular in voice based machine control, [1,2,4], where it is an innate feature [3]. Because voice signal consists of natural language sentences which is inherently composed of imprecise words. Furthermore, in some implementations, speech recognizer is designed to identify a particular set of words which describe the machine functions where users restrain from natural conversation [1,4,5]. Therefore we are encouraged to design an FNN running on unrestricted speech, which can anticipate the above difficulties. The proposed system used a keyword spotting system to identify the machine sensitive words [6–9] and interpret them for machine using FNN. The new concept called "significance of words" is discussed to equate the system output to the users desire.

In the reminder of this paper, in Section 2, the system overview is briefed with its major components: the speech recognizer (SR), action selection network (ASN), action modification network (AMN) and significance of the imprecise words with their characteristics. We discuss experimental setup and results in Section 3 and give conclusions and future directions in Section 4.

2 System Overview

The system functionality should cater for the major demands, i.e., identify the machine sensitive words from the running speech and interpret them to the machine in its identifiable form. Consequently, the system should be capable to give a response similar to user's desire, which should be the output of any control system, contained in the speech signal.

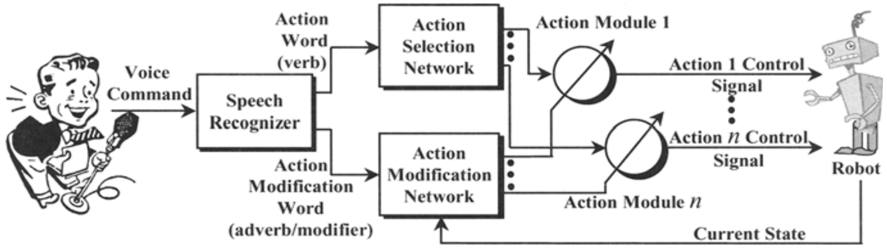

Fig. 2. The system overview

As illustrated in Fig. 2, an SR captures the user's utterances by means of a microphone. Captured voice signal is processed to recognize the machine sensitive words (pseudo sentence) and pass this pseudo sentence to an artificial neural network (ANN) to decode it into an action word (the verb) and several action modification words (the adverb). The nature of the action word is definite or precise but the action modification words are not definite, or imprecise [5]. Therefore, the system handles them in a different manner. The action word is fed into Action Selection Network (ASN) to fire the prospective action. Action modification words together with current machine status are fed into Action Modification Network (AMN), which modifies the operating behavior of the action fired by the ASN. Actions are implemented as modules. The robot may have N different functionalities; as an example, turning capability where it can be activated by turning module, and moving capability where it can be activated by moving module, etc. Each module emits the activation signals for the robot. The ASN fires one of these modules at a time.

2.1 Speech recognizer

The SR is developed using HMM Toolkit[1], which is an integrated suite of software designed for building and manipulating continuous density HMMs to develop speech recognition systems [10].

The SR consists of two parts as shown in Fig. 3. The keyword spotting module has two parallel networks as in Fig. 4. The outline of the work carried out by the keyword spotter can be explained by means of the following conversation:

User	Robot, Can you **go very fast**
Robot	You ask me to **go very fast**
User	Yes/No

[1] Hidden Markov Model Toolkit used in this design was developed at the Speech Vision and Robotics Group of the Cambridge University Engineering Department.

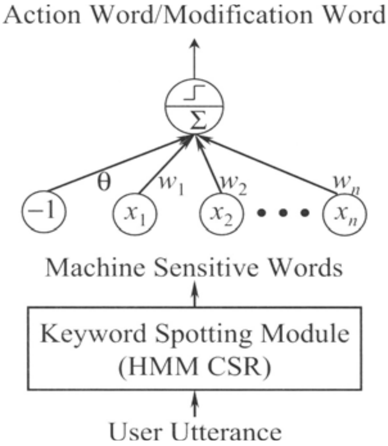

Fig. 3. The speech recognizer

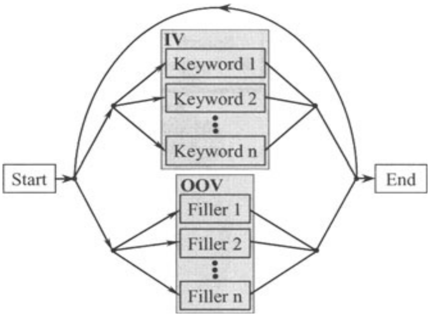

Fig. 4. The architecture of the keyword spotter

Keyword spotter filters the machine sensitive words "**go very fast**" from the utterance. This has been achieved by implementing filler network to identify the out-of-vocabulary (OOV) words as an alternative network to the baseline speech recognition (in-vocabulary or IV) network as in Fig. 4.

Once the utterance has completed, the HMM continuous speech recognizer (HMM CSR) recognizes the lexical contents. Then, the built-in keyword spotter separates out the pseudo sentence and forwards it into ANNs for the identification and classification [11]. Each perceptron identifies a particular word. If the word exists in the user command, output of the perceptron is set to one. Otherwise it is set to zero. The convergence procedure of the perceptron is explained below by using the notations used in Fig. 3. In learning phase, initial weights $w_i\left(t\right), \left(0 \leq i \leq n\right)$, and the threshold at the output node θ, are initialized to small random values. Here $w_i\left(t\right)$ is the weight from input i at time t. Then new continuous valued inputs $x_1,\ x_2,\ \ldots,\ x_n$ along with the desired outputs $d(t)$ are presented to compute the output as

$$y\left(t\right) = f_h\left(\sum_{i=1}^{n} w_i\left(t\right) x_i\left(t\right) - \theta\right) \tag{1}$$

where

$$f_h\left(\alpha\right) = \begin{cases} +1 & \text{if } 0 \leq \alpha \\ -1 & \text{otherwise.} \end{cases}$$

The weights are updated by

$$w_i\left(t+1\right) = w_i\left(t\right) + \eta\left[d\left(t\right) - y\left(t\right)\right] x_i\left(t\right) \tag{2}$$

with

$$d\left(t\right) = \begin{cases} +1 \text{ if input is a desired word} \\ -1 \text{ otherwise} \end{cases}$$

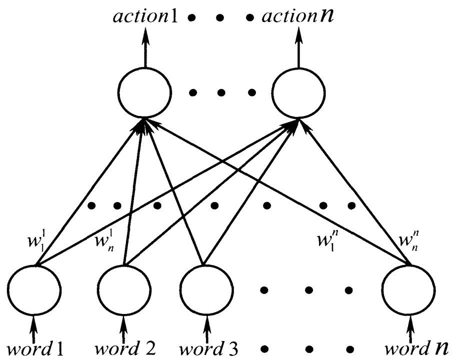

Fig. 5. The action selection network

where η is a positive gain fraction less than 1. Weights are unchanged if the net makes the correct decision.

2.2 Action selection network

Action words recognized at the SR are fed into the ASN illustrated in Fig. 5, to fire the desired action. It also generates a binary output as SR, which switches on the desired output module to trigger the desired action. Namely, if user wants to turn the mobile robot then the turning module is triggered by suppressing the other and vice versa. The ASN is also an ANN consisting of perceptrons, where it is trained using the same methodology explained under the ANN at SR.

2.3 Action modification network

Inherent properties of FNN controllers, i.e. their ability to manipulate imprecise data, naturally persuade us to select it as the controller for designing a machine control system driven by NL [12]. The system proposed here uses the linguistic rules of fuzzy algorithms to seize the user's desire coming in the form of adverbs/fuzzy predicates of the NL.

The proposed AMN shown in Fig. 6, consists of FNN for each and every actions which have been modified by fuzzy predicates (adverbs). The antecedent part takes the current value of the particular action as well as command input of the user for output value calculation at the consequent part. Every part for the each action is learned separately by the gradient descent algorithm, which modifies the consequent part of the particular action in the FNN. The adaptation process is illustrated in Fig. 7.

As shown in Fig. 6, at the layer 1, i.e., at the linguistic labels, every node computes a membership function of $\mu_{A_i}(x_i)$, where x is a crisp input to the

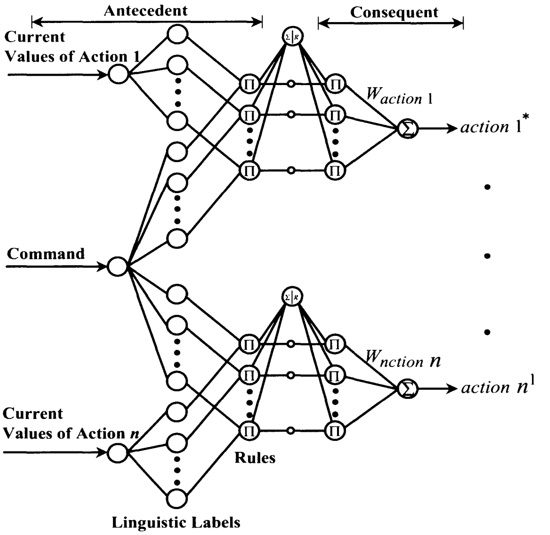

Fig. 6. The action modification network

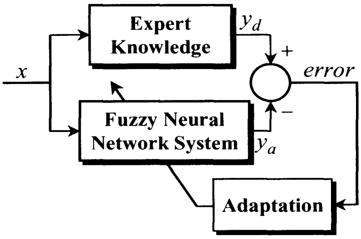

Fig. 7. Scheme for adaptation of AMN

node and A_i is the linguistic label associated with this node. In other words, the first layer is the fuzzification layer and its outputs are the membership values. The shape of these functions is triangular except command nodes. At the each rule node, layer performs the T-norm operator and it is usually the multiplication of the incoming signals:

$$h_i = \mu_{A_i}(x_i) * \mu_{B_i}(y_i)\,. \tag{3}$$

Here h_i is the confidence in the antecedent, $\mu_{A_i}(x_i)$ and $\mu_{B_i}(y_i)$ are the confidences in the linguistic labels, and "$*$" is the algebraic product. Then the output consequent, *action* $k^*(k = 1, ..., n)$, can be calculated as the following weighted mean of w_i with respect to the weight h_i:

$$\textit{action } k^* = \frac{\sum_{i=1}^{r} h_i w_i}{\sum_{j=1}^{r} h_j} \tag{4}$$

where r represents the number of rules.

The connection weights at the consequent part are trained off-line by the information gathered for the particular action. When w_i represents an element of the weight vector W_x, where x means any action, it is updated by using the following equation:

$$w_i\,(t+1) = w_i\,(t) + \gamma\,[y_d - y_a]\,\frac{h_i}{\sum_{j=1}^{r} h_j} \tag{5}$$

where γ represents the learning rate, and y_d and y_a represent the desired output and actual output respectively for the action selected for the training.

2.4 Significance

We came across a new phenomena called as significance here, while interpreting imprecise words for the machine. It describes the contextual meaning of the word, i.e., transformation of words' meaning according to the current state of the machine. In natural language based commanding, this is related with the machine functions, which has limitations. As an example if machine has limited range in velocity in performing its actions the significance can be described as illustrated in Fig. 8. The significance of the command "go very fast" diminishes when machine arrives to its maximum speed, similar effect can be seen with the command "go very slow" at low speeds. This concept is taken into consideration in the adaptation process to represent the user desire more closely in the implementation.

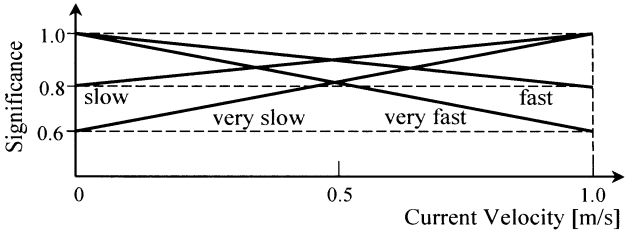

Fig. 8. Significance of fuzzy predicates of velocity command

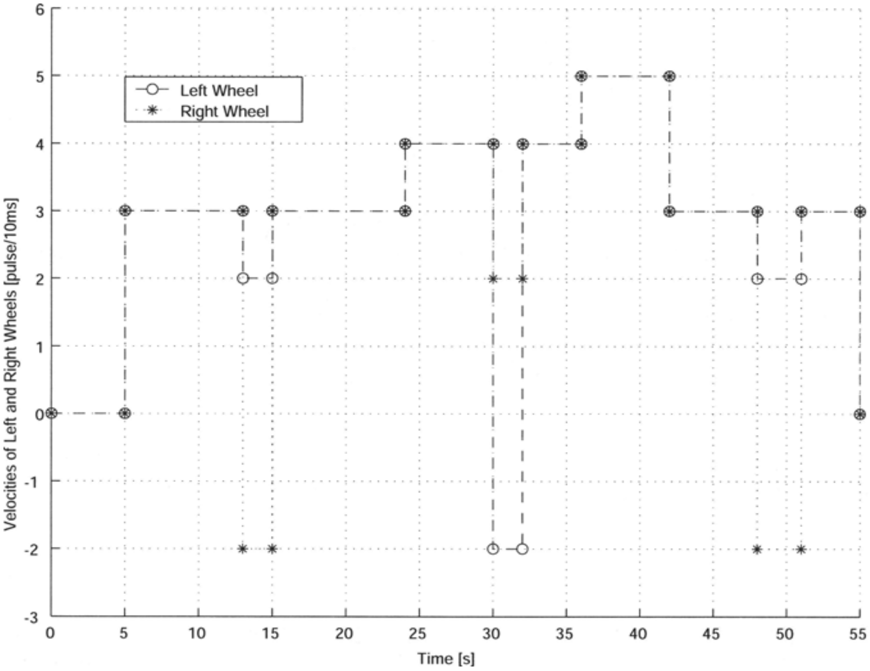

Fig. 9. Velocity profile of the two wheels of the Khepera®

3 Experimental Setup and Results

The concepts described in the above section have been applied to navigate the miniature robot, Khepera®, in real time using the user's vision information transferred through the speech commands, simulating the function of voice controlled wheelchair or navigating a tele-operated mobile robot using vision information captured by a camera. Khepera® used in this experiment can perform two functions moving and turning, which have been used to test the above theoretical concepts in real time. The speech signal captured by the microphone, is processed using the software running on Linux® environment and transferred to the Khepera® by using the RS-232 serial communication protocol. The FNN controller parameters used in this experiment were kept as same as in [5].

At first, the output of the speech recognition module is described here to show the machines capability to capture the machine sensitive words from unrestricted user utterances.

User	robot, can you go very fast
Recognizer	`FILLER FILLER FILLER GO VERY FAST`
User	robot turn right
Recognizer	`FILLER TURN RIGHT`
User	please turn left
Recognizer	`FILLER TURN LEFT`

The above "FILLER" occurrences have been filtered out for further processing. The keywords and out-of-vocabulary (OOV) words used in training are shown in Table 1. The alternative network used in the experiment has one filler node.

The velocity profile shown in Fig. 9, illustrates the velocities of left and right wheels of its 55 [sec] navigation for the commands: "robot, go very fast," "please turn right," "can you go fast," "turn left," "please go very fast," "robot, go very slow," and "please turn right." Velocity values evaluated at the FNN are truncated to nearest integer value since velocity commands to the Khepera® should be in integer format.

Table 1. Keywords and OOV words used for the training of the SR.

Keywords	OOV words
go	please
turn	robot
very	can
little	you
fast	I
left	want
right	
forward	
backward	
to	

4 Conclusions and Future Directions

The two major requirements, i.e., interpreting imprecise words in natural language commands and filtering machine sensitive words from the running utterance, for natural and flexible conversation with machines have been implemented in the experiment. This has immense help in controlling the nature of performing the task assigned by the human counterpart in the cooperation with the robot. Khepera® has two functional capabilities like in wheelchair, but there are no restrictions in using these concepts with the machines having several functional capacities. The design suffers from lack of parsing, where it couldn't identify the context grammar. Hence it couldn't differentiate the commands, "I want to go fast" from "I **do not** want to go fast." We are focusing to include context grammar identification unit in future works. In addition to that we plan to investigate the construction of filler elements for robust elimination of out-of-vocabulary words including background noise and apply these elements to a robot, which has high degrees of freedom.

References

1. Mazo M., Rodrìguez F. J. et al. (1995) Electronic control of a wheelchair guided by voice commands. Control Eng. Practice **3(5)**:665–674
2. Sugisaka M., Fan X. (2001) Control of a welfare life robot guided by voice commands. In: Proc. of the ICCAS 2001, Cheju Korea, 390–393
3. Lin C. T., Kan M. C. (1998) Adaptive fuzzy command acquisition with reinforcement learning. IEEE Transaction on Fuzzy Systems **6(1)**:102–121
4. Komiya K., Morita K. et al. (2000) Guidence of a wheelchair by voice. In: IECON 2000, Nagoya Japan, 102–107
5. Pulasinghe K., Watanabe K. et al. (2001) Modular fuzzy neural controller driven by voice commands. In: Proc. of the ICCAS 2001, Cheju Korea, 194–197
6. Rose R. C., Paul D. B. (1990) A hidden Markov model based keyword recognition system. In: Proc. of the IEEE ICASSP '90, Albuquerque New Mexico, 129–132
7. Jeanrenaud P., Ng K. et al. (1993) Phonetic-based word spotter: Various configurations and application to event spotting. In: Proc. of the EUROSPEECH '93, Berlin Germany, 1057–1060
8. Bazzi I., Glass J. R. (2000) Modeling out-of-vocabulary words for robust speech recognition. In: Proc. of the ICSLP 2000, Beijing China.
9. Leeuwen D. A. V., Kraaij W. et al. (1999) Prediction of keywords spotting performance based on phonemic contents. In: Proc. of the ESCA/ETRW, Cambridge UK, 73–77
10. Young S. J. (1993) The HTK hidden Markov model toolkit: Design and philosophy. Technical Report TR.153, Department of Engineering, Cambridge University UK
11. Lippmann R. P. (1987) An introduction to computing with neural nets. IEEE Magazine on Acoustics, Signal, and Speech Processing **4**:4–22
12. Jang J. S. R., Sun C. T. (1995) Neuro-fuzzy modeling and control. In: Proc. of the IEEE **83(3)**:378–406