Speech Recognitionbased on ROS middleware for Mobile Robots

Master Degree
Information technology
EL MEHDI DIAB
1324316
el.diab@stud.fra-uas.de

Abstract— **Voice speech is a natural manner in which people communicate with mobile robots. First, the speaker normally sends the right voice commands to handle and monitor moving mobile robots. Secondly, a mobile robot with speaking abilities will communicate with people, to reply to the individual or report him for accurate or misconceptions and the execution of commands from human voice. A suitable automated speech model is the popular way to achieve voice communications between individuals and mobile robots. In this study we will discuss about voice recognition and the ability of the system to receive, interpret, analyze, and carry out the spoken commands. Also, implementation of an automatic speech recognition (ASR) model and perform the simulation in ROS middleware, then creating a GUI and executing Voice control mobile robot package in ROS. Furthermore, this paper also discusses our experimental result.**

Keywords—Speech recognition, ROS middleware, ASR.

I. Introduction

Speech recognition plays a vital role in communication and conversation, and individuals with natural hearing skills will listen to varying types of sounds in different acoustic environments at a time. Robots should have an audio capacity equal to that of humans, especially robots which will be of assistance in our everyday environment to reach efficient and seamless human-robot contact. There are other noise sources in everyday settings, including the robot's own motor sounds, in addition to the target speaker source [1].

In this paper is organized as follows. A starting with study of speech recognition methods used in an autonomous robot. In section 3 we develop an ASR model and implemented on ROS package followed by section 4 building GUI and executing the voice control mobile robot package in ROS. Finally, experimental result and analysis is presented as section 5.

II. STUDY OF Speech recognition

Technology of speech recognition aims to change the way we communicate with devices (robots, computers, etc.) in the future. This technology matures every day and scientists continue to work hard to solve the remainder. Nowadays it is introducing many important areas (like - in the field of Aerospace where the training and operational demands on the crew have significantly increased with the proliferation of technology [5], in the Operation Theater as a surgeon's aid to control lights, cameras, pumps and equipment by simple voice commands [3]) in the social context. Speech recognition is the process of converting an acoustic signal, captured by microphone or a telephone, to a set of words [2]. There two important part of in Speech Recognition - i) Recognize the series of sound and ii) Identified the word from the sound. This recognition technique depends also on many parameters - Speaking Mode, Speaking Style, Speaker Enrollment, Size of the Vocabulary, Language Model, Perplexity, Transducer etc [2]. For speech recognition systems there are two forms of Speak mode - one word at one time (isolated speech) and continuous voice. The speech recognition system can also separate - the voice based and the voice independent system dependent on the speaker enrollment. In Speaker dependent systems users must first train the systems and then use them. Speaker dependent systems will, on the other hand, recognise the voice of a speaker. In a speech recognition system the vocabulary and language model are both essential considerations. Word patterns or artificial grammars are used to confine the combination in a sentence or sound chain. The language should also be of a suitable scale. A lot of vocabulary or related sounding terms making the system difficult to recognize. In two decades, Hidden Markov Models was the most common and dominated technique. There are other techniques also use for SR system - Artificial Neural Network (ANN), Back Propagation Algorithm (BPA), Fast Fourier

Transform (FFT), Learn Vector Quantization (LVQ), Neural Network (NN) [4].

III. CREATING OF AN ASR MODEL

To build the speech recognition model, I used The CMUSphinx toolkit is a leading speech recognition toolkit with various tools used to build speech applications. CMUSphinx contains several packages for different tasks and applications. I choose one of the toolkits: Pocketsphinx is a lightweight recognizer library written in C. the internal working of voice recognition in pocket sphinx is divided into: Acquisition, Processing, Recognition, acoustic model and Language model.

A. *Pocketsphinx*

Pocket sphinx is a developing voice recognition software used in embedded systems and hand-held systems. It is derived from sphinx-2 voice recognition software. Pocket Sphinx associates arithmetic algorithms for GMM Computation.

Versions:

Pocket Sphinx 0.2: First version of Pocket Sphinx. This includes support for language model which is defined as a Finite State Grammar (FSG).

Pocket Sphinx 0.2.1: This is a bug-fix release to make fixed point computation work, which was inadvertently broken in the 0.2 release.

Pocket Sphinx 0.2.1*:* This version is accurate and 20% faster than the previous versions. Large amount of code is eliminated thus predominantly making this version smaller.

Working of Pocket Sphinx:

Internal working of Voice recognition in Pocket Sphinx is explained below:

Acquisition: Voice or audio input is segregated into online and offline modes. Here, in this paper online mode is used. Online mode refers to unknown size of data given as input and the recognizer takes it as a small bit of information and processes it. In offline mode, the data is already available making the recognition more accurate.

Processing: At a given instant of time the input audio is represented by real numbers as per the audio volume. Here the precision format is single precision 16-byte little-endian format. Basically, the sampling frequency is 16Khz.

Recognition: Based on the structure of the audio input, the recognition is done by 3 models.

This project requires only two models which are mentioned below:

- Build a dictionary:

A dictionary contains all words that the robot needs to detect, together with the corresponding sequences of phonemes. An acoustic model describes how likely an acoustic realization is given that the text is known.

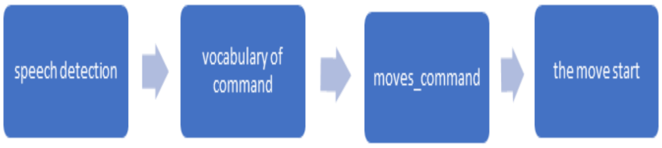
- Build a language model:

It helps to confine the match process by stripping words that are not probable. Accuracy rate increases when the language model is successful in search pace restriction which means that it should be good at predicting the next word.

B. *ROS integration*

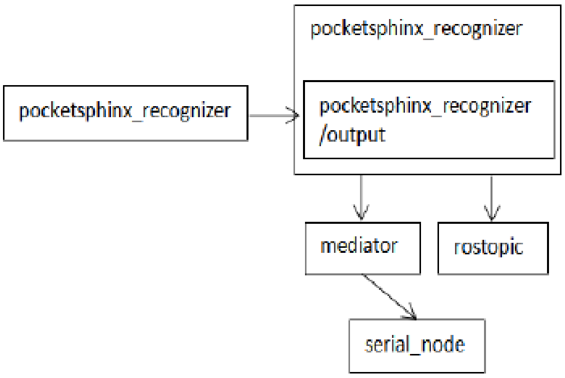
To perform the simulation on ROS middleware I used ROS indigo Python Platform can support the voice control operation use the pocketphinx.

Figure 1 : ROS structure for mobile control



First, I create ROS Package diab_el in my workspace which depends on pocketsphinx, roscpp, rospy, sound play and std_msgs then I edit the dependencies on package.xml.

Figure 2 : Node Connection of the system



on the second step I create my vocabulary of Command or corpus as it is specified in the pocketSphinx in order to control the move where it contains the language model (.lm) and dictionaries (.dic) then I build as an input parameter reognizer.py node till ROS can be able to find the parameters as it is mentioned bellow :

Figure 3 : language model (move commands)

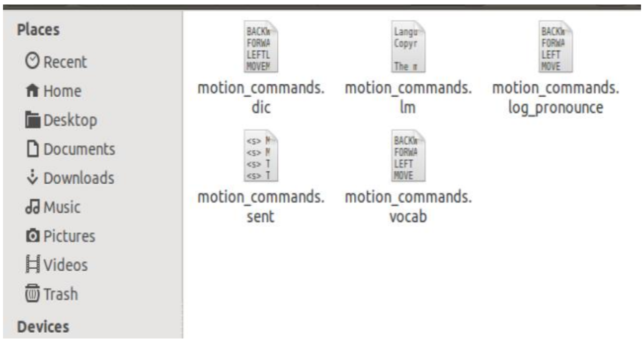
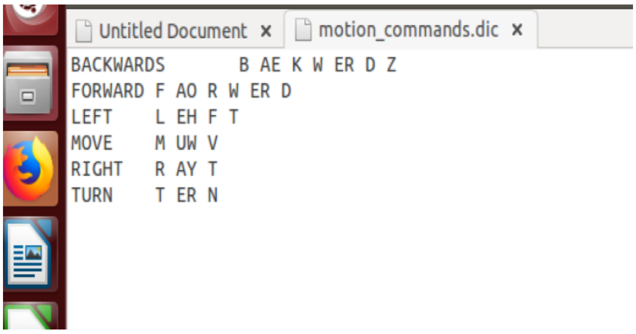


Figure 4 : dictionary doc

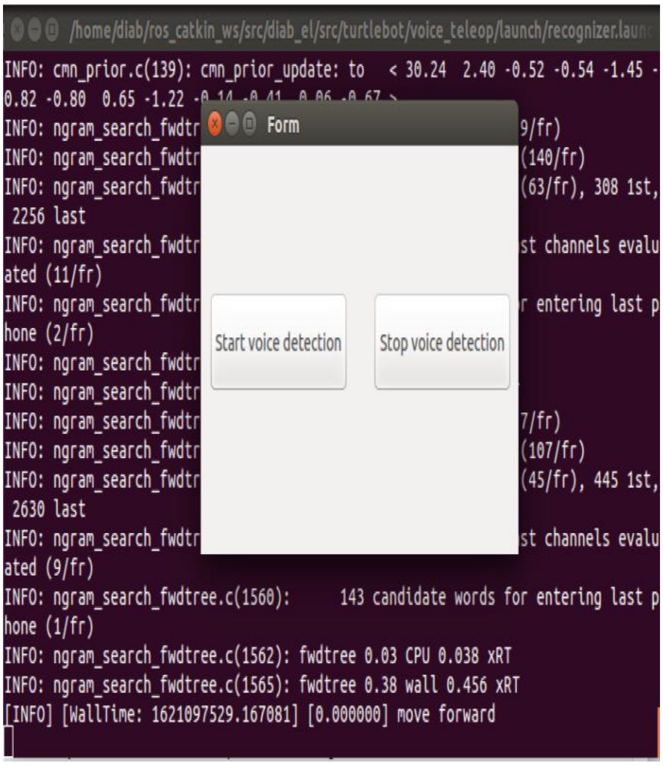


```
BACKWARDS        B AE K W ER D Z
FORWARD F AO R W ER D
LEFT     L EH F T
MOVE     M UW V
RIGHT    R AY T
TURN     T ER N
```

IV. BUILDING GUI

the Graphical User Interface (GUI) is responsible for interacting with our speech recognition model. Using GUI will facilitate the hole communication between the users and the robots rather than trying to understand the complex command to control the robot.

Figure 5 : window of GUI


/home/diab/ros_catkin_ws/src/diab_el/src/turtlebot/voice_teleop/launch/recognizer.laun
Form
Start voice detection
Stop voice detection

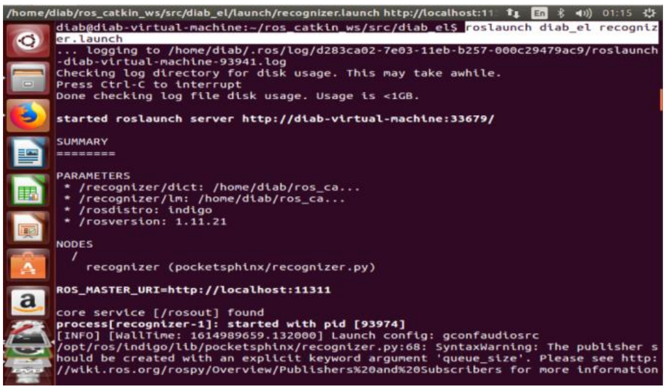
```
INFO: cmn_prior.c(139): cmn_prior_update: to   < 30.24  2.40 -0.52 -0.54 -1.45 -
0.82 -0.80  0.65 -1.22 -0.14 -0.41  0.06 -0.67 >
INFO: ngram_search_fwdtr                                 9/fr)
INFO: ngram_search_fwdtr                                 (140/fr)
INFO: ngram_search_fwdtr                                 (63/fr), 308 1st,
 2256 last
INFO: ngram_search_fwdtr                                 st channels evalu
ated (11/fr)
INFO: ngram_search_fwdtr                                 r entering last p
hone (2/fr)
INFO: ngram_search_fwdtr
INFO: ngram_search_fwdtr
INFO: ngram_search_fwdtr                                 7/fr)
INFO: ngram_search_fwdtr                                 (107/fr)
INFO: ngram_search_fwdtr                                 (45/fr), 445 1st,
 2630 last
INFO: ngram_search_fwdtr                                 st channels evalu
ated (9/fr)
INFO: ngram_search_fwdtree.c(1560):      143 candidate words for entering last p
hone (1/fr)
INFO: ngram_search_fwdtree.c(1562): fwdtree 0.03 CPU 0.038 xRT
INFO: ngram_search_fwdtree.c(1565): fwdtree 0.38 wall 0.456 xRT
[INFO] [WallTime: 1621097529.167081] [0.000000] move forward
```

In the GUI, two tabs are created to execute the voice control mobile robot in ROS:

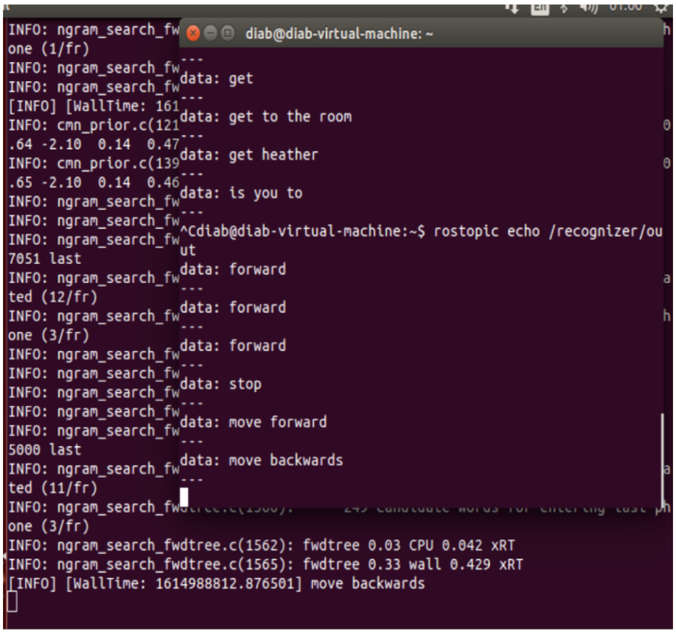
- Start voice detection: in Initialize the system's voice recognition feature and prepare the speaker.
- Stop voice detection: stop the recording voice if the speaker wants to end the communication.

V. EXPERIMENTAL RESULT

Final Step, the experiment that I launched via the recognizer.py under ROS package diab_el and in other terminal I can see the output after giving the robot the specified commands:


/home/diab/ros_catkin_ws/src/diab_el/launch/recognizer.launch http://localhost:11
En
01:15

```
diab@diab-virtual-machine:~/ros_catkin_ws/src/diab_el$ roslaunch diab_el recogniz
er.launch
... logging to /home/diab/.ros/log/d283ca02-7e03-11eb-b257-000c29479ac9/roslaunch
-diab-virtual-machine-93941.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://diab-virtual-machine:33679/

SUMMARY
========

PARAMETERS
 * /recognizer/dict: /home/diab/ros_ca...
 * /recognizer/lm: /home/diab/ros_ca...
 * /rosdistro: indigo
 * /rosversion: 1.11.21

NODES
  /
    recognizer (pocketsphinx/recognizer.py)

ROS_MASTER_URI=http://localhost:11311

core service [/rosout] found
process[recognizer-1]: started with pid [93974]
[INFO] [WallTime: 1614989659.132000] Launch config: gconfaudiosrc
/opt/ros/indigo/lib/pocketsphinx/recognizer.py:68: SyntaxWarning: The publisher s
hould be created with an explicit keyword argument 'queue_size'. Please see http:
//wiki.ros.org/rospy/Overview/Publishers%20and%20Subscribers for more information
```



```
INFO: ngram_search_fw   diab@diab-virtual-machine: ~                          h
one (1/fr)           ---
INFO: ngram_search_fwdata: get
INFO: ngram_search_fw---
[INFO] [WallTime: 161data: get to the room                                    0
INFO: cmn_prior.c(121---
.64 -2.10  0.14  0.47data: get heather                                        0
INFO: cmn_prior.c(139---
.65 -2.10  0.14  0.46data: is you to
INFO: ngram_search_fw---
INFO: ngram_search_fw^Cdiab@diab-virtual-machine:~$ rostopic echo /recognizer/ou
INFO: ngram_search_fwut
7051 last            data: forward                                            a
INFO: ngram_search_fw---
ted (12/fr)          data: forward                                            h
INFO: ngram_search_fw---
one (3/fr)           data: forward
INFO: ngram_search_fw---
INFO: ngram_search_fwdata: stop
INFO: ngram_search_fw---
INFO: ngram_search_fwdata: move forward
INFO: ngram_search_fw---
5000 last            data: move backwards                                     a
INFO: ngram_search_fw---
ted (11/fr)
INFO: ngram_search_fw
one (3/fr)
INFO: ngram_search_fwdtree.c(1562): fwdtree 0.03 CPU 0.042 xRT
INFO: ngram_search_fwdtree.c(1565): fwdtree 0.33 wall 0.429 xRT
[INFO] [WallTime: 1614988812.876501] move backwards
```

VI. Conclusion

In this article, a mobile robot is installed in a voice control system. The voice device demonstrates improved performance, understands the user's voice and acts on the commands of the user. In a real time environment the system has shown to be successful. Manual effort is reduced and no previous preparation for the people to use this equipment is needed. Comprising more reliable, accurate and time saving, the use of speech recognition software through ROS is comparable with android voice recognition applications. This study provides a short insight into the internal functioning of speech systems. Although the device operates in a noisy environment, the provided voice command is likely to be misunderstood.

References

[1] B. C. (2001), Emotive Qualities in Robot Speech., in IEEE/RSJ IROS-2001, , 1389–1394..

[2] S. o. t. s. o. t. a. i. h. l. technology., "Cambridge University Press ISBN 0-521-59277-1,," . Sponsored by the National Science Foundation and European Union, Additional support was provided by: Center for Spoken Language Understanding, Oregon Graduate Institute, USA and University of Pisa, 1996.

[3] N. H. Register.com., "Register Scienece Editor Abram Katz. Operating room computers obey voice commands.," 27 December 2001,. [Online]. Available: http://www.europe.stryker.com/i-suite/de/new haven - yale.pdf (visited 2005-08-15).

[4] M. C. a. R. Sitte., "Analysis of speech reconition thechiques for use in a non-speech sound recognition system.," [Online]. Available: http://www.elec.uow.edu.au/staff/wysocki/dspcs/papers/004.pdf (visited 2005-07-11).

[5] J. Payette, "dvanced human-computer interface and voice processing applications in space.," Plainsboro, New Jersey, , St-Hubert, Quebec, J3Y 8Y9,, March 8-11, p. pages 416–420.