International Journal of Engineering and Technical Research (IJETR)
ISSN: 2321-0869, Volume-2, Issue-5, May 2014

Robot Control with Speech Recognition

Shirish Sharma, Asst. Prof. Sukhwinder Singh

***Abstract*— Speech is the most important way of communication for people. Using the speech as an interface for processes has become more important with the improvements in artificial intelligence. In this project it is implemented to control a robot with speech comments.**
Speech commands were taken to the computer by a microphone, the features were extracted and recognized with Microsoft Visual Studio 2008 (based on C#) which is an integrated development environment (IDE) from Microsoft. It is basically used to develop console and graphical user interface applications. Finally the comments were converted to the form which the robot can recognize and thus, move accordingly using the Arduino Software which is an open source electronics prototyping platform based on flexible, easy to use hardware and software.

***Index Terms*— Speech recognition, Arduino Uno, robot control, SAPI**

I. Introduction

It has always been a dream of human beings to create machines that behave like humans. Recognizing the speech and responding accordingly is an important part of this dream. With the improvements of the technology and researches on artificial intelligence, this dream comes true relatively.
The following is a review paper on a project of robot control with speech recognition that utilizes Arduino Uno, a microcontroller board based on the ATmega328, wherein the speech commands were taken to the computer by a microphone, the features were extracted and recognized with Microsoft Visual Studio 2008 (based on C#) which is an integrated development environment (IDE) from Microsoft.
Controlling the machines and environment with speech makes human life easier and more comfortable. A robot controlled by voice commands. Voice command is taken through a microphone, processed in computer and sent to the robot and finally the robot acts accordingly.

II. Speech Recognition

Speech recognition is the process by which a computer (or other type of machine) identifies spoken words. Basically, it means talking to your computer, and having it correctly recognize what you are saying. Speech recognition in a general environment depends on the following factors:

- Utterance
- Speaker Dependence
- Vocabularies
- Accuracy
- Training

Some speech recognizers have the ability to adapt to a speaker. When the system has this ability, it may allow training to take place. A speech recognition system is trained by having the speaker repeat standard or common phrases and adjusting its comparison algorithms to match that particular speaker. Training a recognizer usually improves its accuracy.
Training can also be used by speakers that have difficulty in speaking, or pronouncing certain words. As long as the speaker can consistently repeat an utterance, speech recognition systems with training should be able to adapt.

Speech recognition systems can be separated in several different classes by describing what types of utterances they have the ability to recognize. These classes are based on the fact that one of the difficulties of speech recognition is the ability to determine when a speaker starts and finishes an utterance. Most packages can fit into more than one class, depending on which mode they're using. The types of speech recognition can be divided into the following:

- Isolated Words
- Connected Words
- Continuous Speech
- Spontaneous Speech

III. Hardware Dependency

Different types of microphone have different ways of converting energy but they all share one thing in common: the diaphragm. This is a thin piece of material (such as paper, plastic or aluminium) which vibrates when it is struck by sound waves. In a typical hand-held mic like the one below, the diaphragm is located in the head of the microphone
When the diaphragm vibrates, it causes other components in the microphone to vibrate. These vibrations are converted into an electrical current which becomes the audio signal.
The electrical current generated by a microphone is very small. Referred to as *mic level*, this signal is typically measured in millivolts. Before it can be used for anything serious the signal needs to be amplified, usually to line level (typically 0.5 -2V). Being a stronger and more robust signal, line level is the standard signal strength used by audio

Manuscript received May 10, 2014.

Shirish Sharma, Student, Department of Electronics and Communication Engineering, PEC University of Technology, Chandigarh, India
Sukhwinder Singh, Mentor, Department of Electronics and Communication Engineering, PEC University of Technology, Chandigarh, India

processing equipment and common domestic equipment such as CD players, tape machines, VCRs, etc.
This amplification is achieved in one or more of the following ways:

- Some microphones have tiny built-in amplifiers which boost the signal to a high mic level or line level.
- The mic can be fed through a small boosting amplifier, often called a line amp.
- Sound mixers have small amplifiers in each channel. Attenuators can accommodate mics of varying levels and adjust them all to an even line level.

IV. Technology

Input text, optionally enriched by tags that control prosody or other characteristics, enters the front-end where a text analysis module detects the document structure (in terms of, e.g., lists vs. running text, paragraph breaks, sentence breaks, etc.), followed by text normalization (expansion to literal word tokens, encompassing transcription of acronyms, abbreviations, currency, dates, times, URLs, etc.), and further linguistic analysis that enables other tasks down the line. The tagged text then enters a phonetic analysis module that performs homograph disambiguation, and grapheme-to-phoneme conversion. The latter process is also called ‘‘letter-to-sound’’ conversion. The string of tagged phones enters a prosodic analysis module that determines pitch, duration (and amplitude) targets for each phone. Finally, the string of symbols that was derived from a given input sentence is passed on to the speech synthesis module where it controls the voice rendering that corresponds to the input text.
Effective evaluation and interpretation of TTS systems is very important. The following parameters need to be taken into consideration for proper assessment of TTS system quality:

A. Naturalness:

The TTS system whose output is closest to human speech is considered to be better one as compared to others. The listener should feel like talking to a human being.
A.Front End Processing: It includes the ability of the system to deal intelligently with commonly used challenges in text such as abbreviations, numerical sequences, homographs and so on.
Examples:
“That cost $5M”.
“I’ll record the record”.
“There were 617,428 callers to (617) 428-4444”

B. Diagnostic Rhyme Test:

DRT is a test of the intelligibility of word-initial consonants. Subjects are played pairs of words with a different first consonant, and asked to identify which word they heard (e.g., dense vs. tense). The system that performs best is the one that elicits the lowest error rate.’

C. Transcription Test:

TTS system can be asked transcribe sentences that have no inherent meaning or context, and therefore minimize the possibility of deriving phonetic information from any source but the speech signal itself, e.g., *“Green ideas sleep furiously”*
If such systems are understood well, it means the system is a good one.

V. The Movement Mechanism

The PWM output pins of the microcontroller are used to provide a varying voltage to control the motors. This is achieved by using the analogWrite() function in the arduino software, which is discussed in later sections. Basically, it converts the maximum output i.e. 5 volts into 256 steps and hence a number in the range 0-255 is given as an input to the analogWrite() function.

The movement of the robot can be divided into four parts-forward, backward, left and right. All the rest of the features can be developed from the above four commands. In case of forward and backward, both the motors are rotated in the same direction. For right and left, one motor is rotated opposite to the other such that we get a on-the-spot rotating motion.

Controlling the direction of the motion is easy. The motor has two inputs, and will rotate if current flows through a potential difference. Applying high voltage to one input and 0 volts to the other starts the motor. Reversing the direction can be achieved by interchanging the inputs. When both inputs are high or low, the motor stops rotating. This feature is basically a braking system for the robot.

VI. Robot Commands

There are fixed commands which can be used to control basic robot actions:

- **Forward** : It is the command to move the robot straight forward unless a new command is sent.
- **Backward** : It is the command to move the robot straight backward unless a new command is sent.
- **Right** : It is the command which turns the robot right unless a new command is sent.
- **Left** : It is the command which turns the robot left unless a new command is sent.
- **Stop** : It is the command which stop the robot.
- **Forty Five Clockwise** : It is the command which turns the Bot 45 degrees in clockwise direction.
- **Forty Five Anticlockwise** : It is the command which turns the Bot 45 degrees in AntiClockwise Direction.
- **135 Clockwise** : It is the command which turns the Bot 135 degrees in clockwise direction.
- **135 Anticlockwise** : It is the command which turns the Bot 135 degrees in anticlockwise direction.

- **Reverse** : It is the command which turns the Bot 180 degrees.
- **Gear Up** : It is the command which increases the speed of the Bot upto Four different Levels .
- **Gear Down** : It is the command which decreases the speed by Four different Levels.

VII. ARDUINO UNO

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.
The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform.

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).
A SoftwareSerial library allows for serial communication on any of the Uno's digital pins.
The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

VIII. PUBLICATION PRINCIPLES

The Speech Application Programming Interface or SAPI is an API developed by Microsoft to allow the use of speech recognition and speech
synthesis within Windows applications. To date, a number of versions of the API have been released, which have shipped either as part of a Speech SDK, or as part of the Windows OS itself. Applications that use SAPI include Microsoft Office, Microsoft Agent and Microsoft Speech Server.
In general all versions of the API have been designed such that a software developer can write an application to perform speech recognition and synthesis by using a standard set of interfaces, accessible from a variety of programming languages. In addition, it is possible for a 3rd-party company to produce their own Speech Recognition and Text-To-Speech engines or adapt existing engines to work with SAPI. In principle, as long as these engines conform to the defined interfaces they can be used instead of the Microsoft-supplied engines.
In general the Speech API is a freely redistributable component which can be shipped with any Windows application that wishes to use speech technology. Many versions (although not all) of the speech recognition and synthesis engines are also freely redistributable.
There have been two main 'families' of the Microsoft Speech API. SAPI versions 1 through 4 are all similar to each other, with extra features in each newer version. SAPI 5 however was a completely new interface, released in 2000. Since then several sub-versions of this API have been released.
The Speech API can be viewed as an interface or piece of middleware which sits between applications and speech engines (recognition and synthesis). In SAPI versions 1 to 4, applications could directly communicate with engines. The API included an abstract interface definition which applications and engines conformed to. Applications could also use simplified higher-level objects rather than directly call methods on the engines.
In SAPI 5 however, applications and engines do not directly communicate with each other. Instead each talk to a runtime component (sapi.dll). There is an API implemented by this component which applications use, and another set of interfaces for engines.
Typically in SAPI 5 applications issue calls through the API (for example to load a recognition grammar; start recognition; or provide text to be synthesized). The sapi.dll runtime component interprets these commands and processes them, where necessary calling on the engine through the engine interfaces (for example, the loading of a grammar from a file is done in the runtime, but then the grammar data is passed to the recognition engine to actually use in recognition). The recognition and synthesis engines also generate events while processing (for example, to indicate an utterance has been recognized or to indicate word boundaries in the synthesized speech). These pass in the reverse direction, from the engines, through the runtime dll, and on to an event sink in the application.
SAPI 5.1: This version shipped in late 2001 as part of the Speech SDK version 5.1. Automation-compliant interfaces were added to the API to allow use from Visual Basic, scripting languages such as JScript, and managed code. This version of the API and TTS engines was shipped in Windows XP. Windows XP Tablet PC Edition and Office 2003 also include this version, but with a substantially improved version 6 recognition engine and Traditional Chinese.

IX. Experimentation and Results

For speech synthesis experimentation, SAPI based application and Speakonia were compared. For their comparison, various features of both the software were compared using different sentences for speech synthesis. For example: A good TTS system should show significant differences in the following sentences:

- Let us pray ; Lettuce spray
- Meet her at the end of Main Street; Meter at the end of Main Street.
- Is the baby crying ; Is the bay bee crying
- It is easy to recognize speech; It is easy to wreck a nice beach.

The two applications showed differences in the text to speech of following:

- TTS of the sentence : "I'll record the record"
- Using 'exclamation mark' in TTS
- Using 'newline' in TTS
- Recognition of area code in the sentence : "There were 617,428 callers to (617) 428-4444"
- TTS of the sentence : " dense and tense"

X. Future Scope

Wireless: Wireless Communication between the BOT and Voice Recognition Mechanism can be developed instead of wired communication.

Industrial Communication: Recent advances in technology now allow industrial robots to perform more and different applications than ever before. Some of these changes like advent of multiple robot control and specifics application robots which in this case could be warehouse management system robot ,plus advances in vision guided technology, connectivity enhancements and improved laser seam tracking and weld inspection.

Other Bot Applications: The various applications like Maze Solving , Tracking , Line Following could be made to done through the further advancements in this Robot System.

Extension of Research Field: Provided that reliable, high-speed transmission is available, there are various possibilities for achieving more comfortable and higher-quality communications. The final goal is to achieve high-quality human-to-machine communications in various environments.

Standardization and Alliances: Information systems have become highly complex with huge variations from one system to another. Generally speaking, it is therefore very important for users, manufacturers, and service providers to work together to establish international standards for interoperability and long-term maintenance. Communications systems can be useful only if users or potential users are attracted to them for their convenience and reasonable cost. For this purpose, we will make the necessary efforts to establish standards and support commercialization. For success in these activities, we will need global collaborations or alliances among organizations and companies, keeping in mind the idea that technologies are good only when users can enjoy them.

Listening Hardware: The average quality of listening hardware has risen, but we tend to listen to music more and more on rather inadequate sources and/or situations: most of music is today being listened to with headphones, loudly and in noisy environments or while driving a car. Intensive research is required for further improvement in this field.

Aid to Handicapped Person: Voice handicaps originate in mental or motor/sensation disorders. Machines can be an invaluable support in the latter case: with the help of an especially designed keyboard and a fast sentence assembling program, synthetic speech can be produced in a few seconds to remedy these impediments.

Vocal Monitoring: In some cases, oral information is more efficient than written messages. The appeal is stronger, while the attention may still focus on other visual sources of information, hence the idea of incorporating speech synthesizers in measurement or control systems.

Multimedia, Man-Machine Communication: In the long run, the development of high quality TTS systems is a necessary step (as is the enhancement of speech recognizers) towards more complete means of communication between men and computers. Multimedia is a first but promising move in this direction.

References

[1] http://arduino.cc/en/Main/arduinoBoardUno
[2] http://msdn.microsoft.com/en-us/library/ms723627(v=vs.85).aspx
[3]http://blogs.msdn.com/b/devschool/archive/2012/02/06/speech-recognition-using-visual-studio-determining-the-bna.aspx
[4]http://stackoverflow.com/questions/1229574/visual-studio-voice-commands
[5]http://visualstudiogallery.msdn.microsoft.com/ce35c120-405a-435b-af2a-52ff24eb2c30