|  |
| --- |
|  |
| ECE478 Project Report |
| Bohr Robot Speech Recognition + Bluetooth |
|  |
| **John Chhokar** |
| **11/15/2010** |

|  |
| --- |
| This document contains an approach to speech recognition and robot control. In addition Bluetooth will be used to call functions from PC to NXT brick. This paper is part of a project for the Niels Bohr Robot at Portland State University. |

**Problem Statement**

To design a speech recognition interface for the Bohr robot. Speech recognition is a complicated task because there are large sources of variability that are associated with voice commands. One type of variability is found in a person’s voice. Acoustic variability’s in ones voice can change the way the computer interprets phonemes. Phonemes are the smallest units of sound and represent 500,000 words in the English language. When the user speaks into the microphone, the user’s voice is converted into digital bytes of information which are comprised of phonemes. The phonemes are then compared to a large library, and when a match occurs the word is then given back to the user. The delivery of speech into the microphone is very critical for this reason.

Another variability associated with voice commands is the user’s environment. External noise associated to an environment can introduce all sorts of problems for the user. For example, a speech program can be fully functional in a conference room where there is no noise, but can be fully not functional when demonstrated in front of a room of noisy kids. For this reason, it’s very important to have a high quality noise cancelling microphone.

Next, it’s important that a speech recognition program is tested in all sorts of environments. This is another crucial step for the reasons stated in the previous paragraph. You can record the accuracy of each command in each environment and predict how successful your program will work when it is demonstrated.

Lastly, I will take two approaches to speak commands to the Bohr Robot. The first approach is to communicate to the Bohr Robot over a serial wired cable and give the robot commands over a wireless noise cancelling headset. Secondly, I would like to add Bluetooth to this robot and control the robot by first giving it a command into the headset and then transmitting that command over Bluetooth.

**The Software**

I am using Microsoft Visual Basic 2008 express edition. In order to include speech recognition into your project you have to include references to the Microsoft Speech Object Library (COM) and System.Speech (.Net). Since my computer is running Windows XP I had to download Microsoft’s Software Development Kit 5.1(SDK 5.1) from Microsoft’s website. Links to download the required software are provided below. All the software I downloaded was free.

Microsoft Software Development Kit 5.1

<http://www.microsoft.com/downloads/en/details.aspx?FamilyID=5e86ec97-40a7-453f-b0ee-6583171b4530&displaylang=en>

Microsoft Visual Basic 2008 Express Edition

<http://www.microsoft.com/express/Downloads/>

**10/31/10 - Speech**

After two weeks of researching speech recognition I implemented a speech recognition algorithm in software. Using Microsoft’s Speech Application Programming Interface (SAPI) a basic algorithm was developed.

**Start**

**Load Grammar**

**Libraries**

**Wait for speech detection**. Move to next state when a word is spoken

**Hypothesis**

Move to next state when word is hypothetically found

**Display Text**

When the program begins, the program waits for speech detection. When a word is spoken into the microphone, the word is broken into phonemes and then goes through Statistical Modeling to try and find the word spoken. An example of modeling used in software is called the Markov Model. An example of how a computer would interpret the word tomato is shown below. The word tomato is broken up into several phoneme’s (T, ow, m, aa, t, ow). If you follow the phoneme through the model below you will see that tomato can be pronounced two different ways but at the last branch of the model there is a 90% chance that the word is tomato. That is the word that is then selected. This type of modeling enables a much quicker type of speech recognition when compared to a brute forced method.



There is one significant drawback to this model. Since this method try’s to match the word spoken into the microphone to several words which may sound the same in the English language the results are not optimal and often times return the wrong word. For this reason I must look at creating my own language library instead of using Microsoft’s library.

One way to improve the speech recognition on a system is to take the speech recognition training on your laptop. The following training is available in your control panel in the speech icon. The more training you take the more accurate your speech recognition becomes.



 

**11/3/10 – Bluetooth**

I wanted to also implement Bluetooth communication so I purchased the following Bluetooth adapter for my computer. This Bluetooth adapter is made by IOGEAR and has a maximum wireless range of 30feet.



**11/4/10 – Speech Demo to Class**

One possible approach to speech recognition was presented to the class. This demonstrated the background to speech recognition and how you can use voice commands to call functions. The demonstration also showed how difficult speech recognition is. I am going to look at ordering a new headset and also look to implement my own grammar library using XML. I need to shoot for 95% accuracy of speech recognition.

**11/5/10 – Bluetooth**

While I am waiting for my headset to come in, I thought I would look at communicating to the NXT controller using my new Bluetooth adapter. My thinking is that I can give the robot simple commands such as forward, reverse, right and left using the headset and that command will be sent via Bluetooth. The previous group had implemented wireless communication using a PSP controller. I wanted to use Bluetooth because it gives you a chance to learn the Bluetooth protocol for communicating between the NXT controller and your laptop. This protocol is similar to software written in ECE371 and ECE372

Bluetooth connection was successfully established between my laptop and the NXT brick. To learn the protocol for Bluetooth and for calling sound files and software files, you have to reference Lego Mindstorms NXT direct commands found in Appendix 2. The link for the document is shown below.

<http://code.google.com/p/smart-robot/downloads/detail?name=Appendix%202-LEGO%20MINDSTORMS%20NXT%20Direct%20commands.pdf>

**11/6/10 - Speech**

To improve speech recognition I bought a pair of high quality noise cancelling headphones from Costco for $89.00. I realized that this was a crucial part for the project and this headset gives me the ability to communicate through USB wirelessly with 3 feet of range.



**11/7/10- Speech**

To improve the speech recognition interface I looked at creating my own grammar library instead of using Microsoft’s grammar library. My initial thinking is that this would improve the program significantly because it would limit the amount of words the engine had to find. I had to learn XML syntax to create my own grammar library. After reviewing examples on XML I was able to successfully implement my own grammar library. Microsoft’s SDK5.1 comes with a grammar library compiler test environment. Here you can write your grammar libraries, compile and run them. If your voice commands are recognized here then you can conclude that the voice commands will be recognized in your program. An example of the grammar compiler is shown below.



**11/9/10 – Robot Hardware**

We need a 12volt battery to power the motors on the base. The battery used by last year’s group is missing. The 12 volt battery shown below was purchased for $30.99.



**11/11/10 – Bluetooth**

I was able call sound files and software files on the NXT brick. I implemented the following functions and drove the robot around the engineering lab using bluetooth: forward, reverse left, right and stop. Next steps are to use speech recognition to drive the robot. The Bluetooth protocol the robot uses is shown below. Thanks go out to my lab group for figuring out the wiring of the motors.

In order to call software files, or sound files, it’s important to really understand the protocol architecture. To call a software file, you have to follow the protocol shown below under the Start Program. I will cover an example for calling a left function on the NXT. The left program would be already programmed into the NXT to turn the robot left. To call the left program in the NXT you have to call left.rxe.





<http://code.google.com/p/smart-robot/downloads/detail?name=Appendix%202-LEGO%20MINDSTORMS%20NXT%20Direct%20commands.pdf>

**Blue Tooth Protocol for calling software files.**

.

Byte 0:How many bytes total in your message. Don’t count bytes 0 and 1.

Byte 1: Always send 0x00 for this byte.

Byte 2: You have two options here. If you send 0x00 then the NXT will send you a return message. If you send 0x80 then no return message is sent. For most of the programs that I wrote I left this byte programmed with 0x00.

Byte 3: This is the command Byte. Sending the byte 0x00 tells the NXT that you are going to call a function.

Byte 4 - 21: In these bytes you send the ASCII equivalents of your file name. For example, byte 4 would be coded with the ASCII equivalent of the first letter in your software file name. In this example we are calling the software file right so ASCII (“l”) is sent.

Byte 5: Here you would send the ASCII value for “e”.

Byte 6: Send ASCII value of “f”

Byte 7: Send ASCII value of “t”

Byte 8: Send ASCII value of “.”

Byte 9: Send ASCII value of “r”

Byte 10: Send ASCII value of “x”

Byte 11: Send ASCII value of “e”

Byte 12: Send null terminator character “0” to end string. This is important!

An example of calling the left program from the NXT is shown below. You can call all of your software files on the NXT by modifying the program code below to match your own software files in visual basic.

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\* This functions calles the left software file on the NXT

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

 Dim byteOut(13) As Byte

 byteOut(0) = &HB '11 bytes in output message

 byteOut(1) = &H0 'should be 0 for NXT

 byteOut(2) = &H80 '&H0 = reply expected &H80 = no reply expected

 byteOut(3) = &H0 'Command

 byteOut(4) = Asc("l") 'l character

 byteOut(5) = Asc("e") 'e character

 byteOut(6) = Asc("f") 'f character

 byteOut(7) = Asc("t") 't character

 byteOut(8) = Asc(".") '. character

 byteOut(9) = Asc("r") 'r character

 byteOut(10) = Asc("x") 'x character

 byteOut(11) = Asc("e") 'e character

 byteOut(12) = &H0

 SerialPort1.Write(byteOut, 0, 13)

**11/20/2010 – Next Meeting**

The plan for this Saturday is to work on writing code for the robot arm. We currently don’t have access to the arm but have a good understanding of its functionality from last year’s group project’s documentation. We are expecting the arrival of the 12 volt DC battery in addition to another NXT brick we will be using for software tests. In addition to adding functionality of the arm we will begin to look at controlling the head of the robot using the VSA software.

I will also look to implement the functions (forward, reverse, left and right) using speech recognition.

**Accomplishments to date:**

* Speech recognition software written and tested. XML library written to improve accuracy. High quality headset ordered.
* Bluetooth setup and controlling robot movement such as forward, reverse, left and right.
* GUI created(See below)

**Next Steps:**

* Control Bohr movements (drive forward, reverse, left and right) with speech.
* Get functionality of robot arm
* Get functionality of robot head
* Integrate speech and Bluetooth with vision being worked on by fellow group member JC Arada.