Autonomous Vehicle Speaker Verification System

Functional Description and System Block Diagram

Group Members:

Aaron Pfalzgraf
&
Christopher Sullivan

Advisor:

Dr. Jose Sanchez

Date:

September 30, 2013

Objective:

This project will involve developing a speaker verification system to assist in reliably controlling a robot with voice commands. Voice commands are a convenient robot control method, but they introduce safety risks if anyone's voice can control the robot. By including a speaker verification system, the robot can be set to respond only to one "master" voice. This will increase the safety and reliability of the system.

Speaker Verification Background:

Speaker verification refers to speech processing systems that can validate the provided identity of a user by examining the sound of his or her voice. For proper system operation, the user must claim to be an individual about which the speaker verification system already has speech information. The system's speech information database must be generated by training sessions before any verification can occur. When a user speaks into the system's microphone, the system will compare the new speech samples with the stored speech information from the pre-generated database. If the speech patterns match within a certain threshold, the user's identity claim is accepted. Otherwise, the claim is rejected. Such systems are valuable and convenient in security applications, because the sound of a user's voice is not something that the user can forget or misplace like a password or card-key. In robot control, speaker verification is useful so that the robot can only be controlled by one person's voice.

Speaker verification should not be confused with *speaker identification*. In speaker identification systems, the user need not claim an identity, and his or her speech is compared to the stored speech information of multiple individuals to determine the closest match. Speaker verification systems are only concerned with confirming or denying a claimed identity. While the claim typically refers to only one person, it could also refer to a group of people. The important distinction is that speaker verification systems are not intended to scan the entire database every time they operate.

While there exist a large number of methods for designing speaker verification systems, such methods can be subdivided into two groups: *text-independent verification* and *text-dependent verification*. Text-dependent systems prompt the user to speak a specific word or phrase during the training session to develop a unique speech profile. During the verification stage, the user is again prompted to speak the same word or phrase to optimize the system's ability to compare the speech samples. Such systems can require shorter speech sampling times to achieve high accuracy, but the potential for the user to misread the prompt contributes to the error margin. Text-independent systems on the other hand do not require the user to speak any specific words or phrases. This can reduce user error and increase convenience, but it also requires longer speech samples and more memory space to store all the additional data. Text-independent systems must also be prepared to process more possible utterances than text-dependent systems. They involve more complicated calculations and are therefore harder to implement in real-time.

Project Overview:

The text-dependent speaker verification system will be implemented with a Texas Instruments 55x series DSP board. This board was chosen for its low-power capabilities. The system will be mounted to the chassis of a simple wheeled robot so that the user can control the robot with a small set of voice commands if the speaker verification system accepts his or her identity. A microcontroller will serve as the control interface between the DSP and the motor. The robot will not follow the instructions of anyone other than its "master" defined during the training session. In this case, the text-dependence of the system is due to the set of pre-defined voice commands that the robot can accept. Fig. 1 shows a block diagram of the physical connections of the system. Fig. 2 shows a high-level flow chart of the speaker verification system to be programmed on the DSP board.

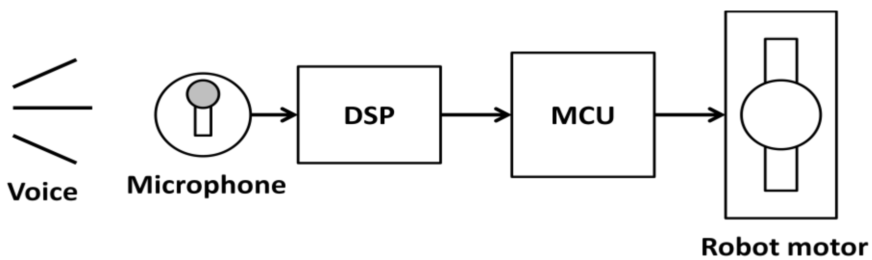

Fig. 1, Block diagram of the hardware connections of the system

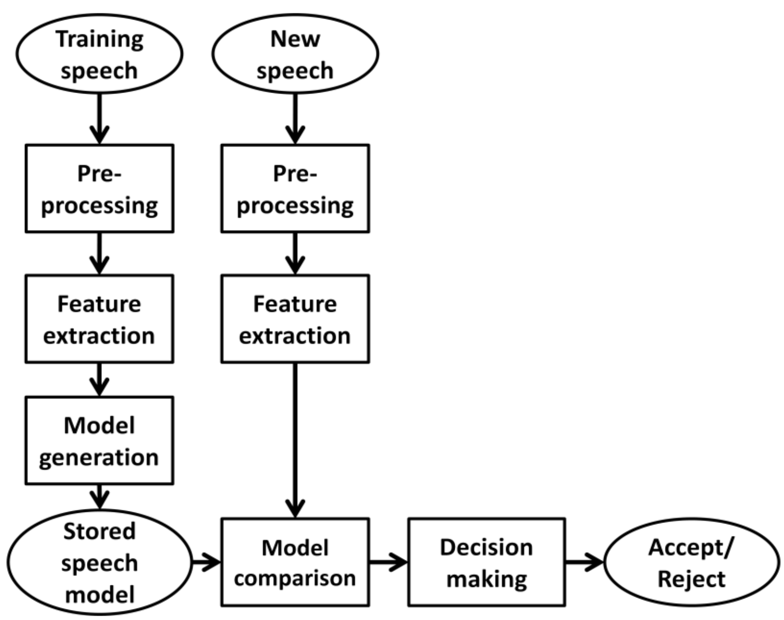

Fig. 2, High-level flow chart of the speaker verification system to be implemented on the DSP

Integrating the speaker verification system into a voice-command-based robot control system will require some special considerations:

- The "master" speech model stored in the database must contain as many entries as there are command words for the robot in order to retain the text-dependent nature of the system.
- A simple speech recognition system must be introduced before the speaker verification system to determine which command to send to the robot and which stored speech model to use during model comparison.
- If the microphone is attached to the robot, the quality of the speech samples will be greatly deteriorated by motor noise and acoustical shifting as it moves through the room. This can be partly mitigated with proper microphone selection, but only conducting tests will determine if these concerns can be ignored.
- If the microphone is not attached to the robot, a reliable and unobtrusive means of connecting the microphone to the DSP board must be found.

This project is first and foremost focused on designing the speaker verification system, so the complications mentioned above will not be addressed until a typical-case speaker verification system is simulated in MATLAB and implemented on the Texas Instruments 55x series DSP board. The implementation sequence will be as follows:

1. Simulate the speaker verification system in MATLAB. Pre-recorded speech samples will be loaded into the simulation to generate a database. Additional pre-recorded speech samples will be used to test the verification abilities of the system.
2. Implement the speaker verification system on the Texas Instruments 55x series DSP on a test-bed. Real time speech from a stationary microphone input will be used to generate the database and perform verification. Verification of multiple command words will be tested, but no robot control will be included.
3. Implement the speaker verification system with the robot. The user should be able to control a robot with simple commands using the speaker verification system from the test-bed implementation and a motor control MCU.

Functional Description:

While many strategies exist for designing speaker verification systems, most follow the high level block diagram shown in Fig. 2.

Data Gathering: This block includes the user's speech production, the microphone, and the text prompt. For this robot control system, the text prompt can be considered as one of a small set of possible command words. The microphone placement, type, and sampling rate will affect the speech samples. The condition of the speaker's voice will also affect these samples. It will be beneficial to design a system that can maintain accuracy even when the speaker has a head cold or is suffering from some other vocal abnormality.

Pre-processing: This block includes band-limiting the input speech to remove unwanted frequencies, cancelling out unwanted noise, removing samples of silence, and performing any other signal manipulations that increase the comprehensibility of the speech input.

Feature Extraction: This block generates vectors of parameters, or features, to describe the speech signal. Features that have been shown to assist with speaker verification include fundamental pitch, harmonic content, linear prediction coefficients, and many others. Processing time, accuracy, and ease of implementation will determine which feature extraction methods are used in this system.

Model Generation: This block generates models that describe the feature vectors of different individuals' training speech signals. These models are stored in the speech information database of the system. Model generation strategies include cluster analysis, vector quantization, neural networks, and many others. Accuracy and ease of implementation will determine which model generation method will be used by this system. Processing time is less important to this block, because the model generation calculations do not need to be completed in real time.

Model Comparison: This block compares the extracted feature vectors from the new speech sample with the preexisting "master" model that was generated in a training session. The method of comparison will differ based on the model type. Assuming the model type is a vector quantization codebook, this block will quantize each input vector to one of the bins. Once the vectors are quantized, the system will compare the location of the vector within the bin to the probability density function of the "master" vectors assigned to this bin in the training session. If the new feature vectors exist in high probability density areas, then the probability score will be high. Otherwise, the score will be low. Summing the score of each vector yields the total score of the new speech signal.

Decision Making: This block compares a score from the model comparison block with a threshold score to either accept or reject the user's identity claim. The threshold must be chosen to minimize false acceptance and false rejection errors based on the needs of the system.

Conclusion:

This project is intended to design a speaker verification system and to experiment with the use of this system in a robot control environment. With a voice-command-based robot control system, it is desirable that the robot only responds to one or a few pre-determined "masters" to avoid hijacking and promote safety. To design a speaker verification system robust enough for robot control applications, various methods of pre-processing, feature extraction, and model generation must be tested. The considerations brought up in the Project Overview section must also be addressed.

References:

I wrote this at home without access to the H: drive. Please add these in. Thanks.