Autonomous Vehicle Speaker Verification System (AVSVS) Project Proposal

Team Members:

Aaron Pfalzgraf

Christopher Sullivan

Project Advisor:

Dr. Jose Sanchez

Bradley University

11/26/2013

Project Summary:

Speaker verification systems are capable of verifying the identity of an individual by the sound of their voice. Such systems are not to be confused with speech recognition systems which are intended to determine what word a user has said regardless of that user's identity. This characteristic of speech recognition systems makes them inherently unsecure for voice-command-based autonomous vehicle control applications. This project seeks to mitigate these security risks by integrating a speaker verification system into a voice-command-based autonomous vehicle control system. The final result will be an autonomous vehicle that will accept voice commands from only one designated operator.

System Overview:

Hardware:

The complete system block diagram is shown in Fig. 1. A speaker talks into the microphone to generate a digital speech signal. This digital speech signal is passed into the digital signal processor. The digital signal processor processes the speech as either training data or verification request data. It generates and stores a new comparison model in the first case or checks the data against an existing comparison model in the second case. If verification request data is adequately similar to the stored designated operator comparison model, the DSP sends the operator's voice command to the microcontroller unit. The MCU performs motor control and makes the autonomous vehicle obey the operator's command.

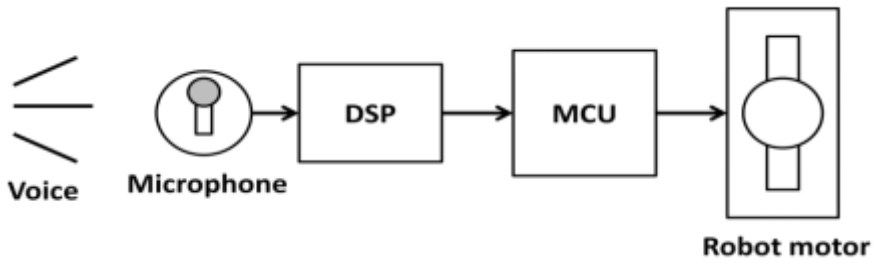

Fig. 1, Hardware connections

Software:

The software block diagram to be implemented on the digital signal processor is shown in Fig. 2. The software follows two distinct paths depending on whether the input data is training data or verification request data. For ease of understanding, this diagram

assumes that model generation is accomplished by the DSP. Model generation may be performed in MATLAB and hardcoded into the DSP.

The speech data is windowed into 20-40 ms long frames using a Hamming window with 50-70% overlap in the pre-processing block. A feature vector of at least 10 mel-warped cepstral coefficients is calculated to describe each frame during feature extraction. These feature vectors are used to train an artificial neural network comparison model if the signal is training data. Otherwise, the feature vectors are passed into the existing artificial neural network comparison model to generate a similarity score between the designated operator's voice and the current user's voice. If this similarity score is above the acceptance threshold, the speaker is accepted as the designated operator, and the DSP communicates his or her voice command to the MCU.

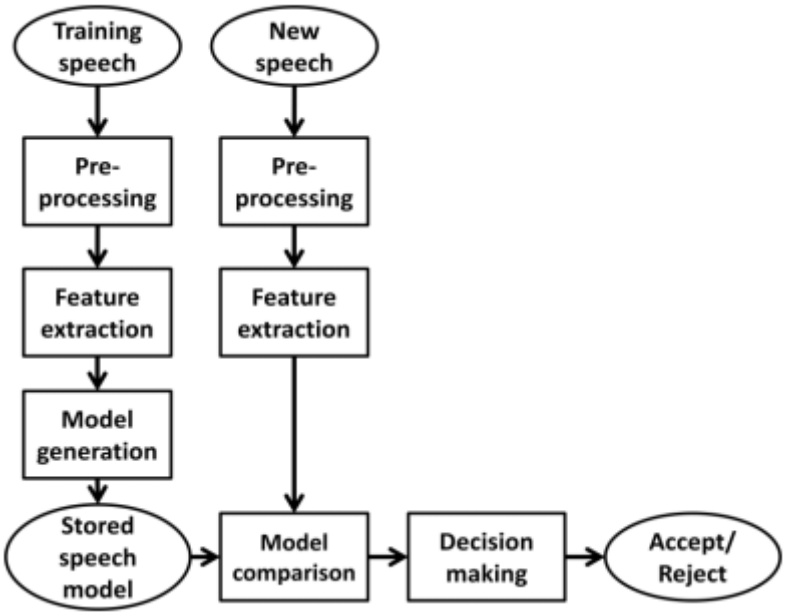

Fig. 2, Software block diagram

Performance Specifications:

- Operator rejection error shall be minimized to under 1% for safety reasons
- Imposter acceptance error is desired to be under 2% but may be modified to accomplish the desired operator rejection error percentage
- Speech shall be sampled at 16 kSamples/sec
- System shall function properly in a mildly noisy environment
- Maximum operator-to-vehicle distance for proper functionality shall be at least 10 ft
- Speaker verification shall be accomplished without delay time perceptible to the operator

Hardware Decisions and Equipment List:

Microphone:

- 180 degree pickup
- Up to 25 ft. range
- USB Powered
- Digital line out for listening

Digital Signal Processor:

- C5505 ezDSP from Texas Instruments
- USB Powered
- Digital line out and in
- Support for I2C
- 150 MHz clock

Initial Simulation Results:

Simulations of the speaker verification system have been carried out in MATLAB. The simulations were focused on training an artificial neural network to differentiate between a certain individual's voice and anyone else's voice. For these tests, all imposters were tested against Cree's voice.

Simulation Conditions:

- Population size of 9: 2 female, 7 male
- Tested the words "stop" and "go"
- Speech recorded with an AKG D5 dynamic microphone using a PreSonus Audiobox pre-amplifier at 16 kSamples/sec
- Artificial neural network trained with 50,000 iterations using back-propagation and adaptive learning
- 2 hidden layers with 20 nodes each in the artificial neural network
- 15 mel-cepstral coefficients calculated per 25 ms speech frame with 70% overlap
- Acceptance score threshold set to 0

Simulation Results:

"Stop":

- 0 true speaker rejection out of 28 generated networks
- 0 imposter acceptances out of 28 generated networks

"Go":

- 1 true speaker rejection out of 28 generated networks
- 0 imposter acceptances out of 28 generated networks

Simulation Discussion:

The conducted simulation shows that the variability among artificial neural networks trained using the same data is almost sufficiently low to meet error specifications. This variability can likely be reduced by editing a few parameters from the simulation conditions. Finding which parameters to edit will require further testing and research. Successfully meeting the desired error specifications will also require generating an artificial neural network with low score variability among many different speech samples of the same user. This characteristic could not yet be tested by simulation, but it will be far easier to test once the system is implemented on the digital signal processor with an attached microphone. Until the microphone can be used to test a huge variety of speech samples quickly, the quality of the system will continue to be evaluated based on the variability among networks generated with the same pre-recorded training data.

Schedule:

The work next semester will be divided into several sections. The primary part of the work will be coding the current system we have in MATLAB into C code using the Code Composer IDE. In order to accomplish this, we allowed ourselves some time to learn Code Composer. The next major task would be to interface MATLAB array with a C array. This is because we plan on doing our model generation in MATLAB and then coding the weights on the DSP. The last major part of the work will be spent organizing the research into papers, presentations, etc.

References:

[1] J. P. Cambell Jr., “Speaker Recognition: A Tutorial”, NSA, Ft. Mead, MD, Sep. 1997.

[2] F. K. Soong et al., “A Vector Quantization Approach to Speaker Recognition”, AT&T, Murray Hill, NJ, 1985.

[3] T. Kinnunen et al., “Comparison of Clustering Algorithms in Speaker Identification”, Univ. of Joensuu, Joensuu, Finland.

[4] A. K. Jaine et al., “Artificial Neural Networks, A Tutorial”, Michigan State University, East Lansing MI, Mar. 1996.

[5] Practical Cryptography, “Mel Frequency Cepstral Coefficient (MFCC) Tutorial”, http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/, Oct. 2013.