18-551, Spring 2002
Group 13, Final Report

MY PANTS ARE ON FIRE!
Automatic Lie Detection Using Voice Stress Analysis

Diane Budzik
Andrew Pomerance
Eric Carr
Patrick Durham

Abstract

The human capacity for lying and the problems caused by such deceit have led to the creation of numerous lie detection techniques, with the first reliably accurate one being the polygraph machine. Although the polygraph has been shown to achieve an accuracy of 90% in the hands of a skilled examiner, it does have constraints. Polygraph tests are time-consuming, costly, and cumbersome to give because of all the monitoring equipment required. In addition, the success rates are highly dependent on the polygrapher's ability to accurately read the charts. Our solution to this problem is a speech-based lie detector that uses voice stress analysis. Research has shown that stress can be detected by examining various features of a person's voice. A reliable speech-based system would be less expensive, more mobile, and depending on the accuracy of the algorithm, an expert to analyze the results may be no longer necessary. Such a portable voice-based lie detection product exists, the Psychological Stress Evaluator (PSE). However, it offers poor reliability. One of the lie detection schemes we implemented, the Hirsch and Wiegele scoring method, offered better results (75% accuracy) by utilizing an enhanced variation of the PSE's detection method. The pitch contour method was implemented but only yielded a 50% accuracy rate. The Teager Energy Operator method of lie detection was explored, and despite promising results, it was never ported to the EVM due to time constraints. Two databases were used for testing - the SUSAS (Speech Under Simulated and Actual Stress) database and a specialized database assembled from three polygraph sessions. To implement the lie detection algorithms, voiced/unvoiced detection was needed as well as cepstral analysis. Some preexisting code was utilized, with the majority of the code being written by the

authors. C code for two versions of the lie detector were written (real-time and offline) with the offline version used in the demo.

I. Introduction

The Problem

Since the dawn of man, the human capacity for lying has been an obstacle in the pursuit of justice. The desire to promote one's interests often prompts people to act dishonestly, and this capacity for deceit has led to the creation of numerous lie detection methods. The techniques utilized for lie detection over the course of history range from the arcane to the pseudo-scientific, although the polygraph machine was the first to offer reliable success rates.

A polygraph test is just that; it monitors several physiological variables during questioning. For example, the participant's respiratory patterns (both abdominal and thoracic), pulse rate, and electrodermal activity (EDA) are often measured. Valves are wrapped around the chest and stomach to measure expansion and contraction for characterizing the respiratory response, the EDA is obtained using finger clamps designed to detect sub-dermal neural activity, and blood pressure readings acquired via an arm cuff provide the pulse rate. Needless to say, one could not discreetly verify claims using such equipment. However, in the hands of a skilled examiner, this equipment can be used with 90% accuracy, with many polygraphers claiming success rates of 97%.

Polygraph readings are time-consuming, costly, highly invasive, and require many pieces of equipment to administer a test. In addition, the results are greatly dependent on the examiner’s ability to read the charts. Lastly, the participant’s state of mind and health (hunger, fatigue, illness, and medication are all polygraph killers) can affect the validity of the measurements.

The Solution

Our project is a response to the drawbacks and constraints of a standard polygraph test and reading. It has been shown that the effects of stress on the body can be detected in a person’s vocal patterns by extracting and examining various features of their voice. For instance, it has been shown that there exists a microtremor in human speech, which disappears in times of stress [2]. According to results from psychology, lying produces stress in a non-psychotic individual and should manifest itself in changes in the individual’s speech patterns. Based on this research, we have attempted to develop a system that can reliably detect lies based solely on so-called voice stress analysis. The advantages of such a system are evident. A speech-based system would be significantly less expensive; a reliable system could be built with a microphone and microprocessor rather than the expensive, specialized equipment needed for polygraph machines. Because no device needs to be attached to the participant, setup would be trivial and tests could be administered over the phone. Depending on the accuracy and sophistication of the detection algorithms, an expert may not even be necessary to analyze the results. As a result, the common man could begin to enjoy the same level of assurance that was previously reserved only for law enforcement agencies and large corporations.

In fact, such a portable voice-based lie detection product exists – the Psychological Stress Evaluator, or PSE. The PSE has been shown to offer variable results, however. One method implemented in this project has better results by utilizing an enhanced variation of the PSE's detection method (Hirsch and Wiegele scoring, or H&W). Additional methods were to be used to increase the potential accuracy of our detector. Unfortunately, one method (pitch contour) produced unreliable results and the other (based on the Teager Energy Operator) could not be implemented due to time constraints.

Ultimately, our project was able to distinguish between the lies and truths of the standard polygraph readings we used as our data sets more than 75% of the time, though not so great as to be comparable to polygraph examinations. As a simple stress detector, based on examination of sound files from the SUSAS (Speech Under Simulated and Actual Stress) database, our method was far more successful, though that may be due in part to the nature of the SUSAS library.

Prior 18-551 Work

Several groups in the past have done speech-based projects; examples include morphing, translation, transcription, karaoke training, vocoding, and idiom detection. This project shares its origins with these, since it incorporates several common speech-processing techniques (such as voiced/unvoiced detection and cepstral analysis). However, the topic of lie/stress detection is new to 551, and the methods implemented in this project are without precedent in previous years.

II. General Speech Processing

Voiced/Unvoiced Detection

All of the methods proposed below work on voiced speech segments; unvoiced speech segments are mostly noise, whereas voiced segments convey the psychological state of the speaker. Thus, accurate classification of voiced/unvoiced segments is key to any processing.

Ahmadi and Spanias propose in [1] three features of a speech segment that can be used to classify a segment as voiced or unvoiced. They compute the energy, cepstral peak, and zero-crossings of the signal and compare these to a threshold. The thresholds they choose are the medians of the respective feature, which are calculated in a separate training session. If any of these features are above the threshold, the segment is declared voiced; otherwise it is declared unvoiced. However, for this project only the energy threshold was used, which produced acceptable results.

Cepstral Analysis

Central to all the stress detection methods is cepstral analysis. One use of the cepstrum is to detect the fundamental frequency (pitch) of a voice signal. This property is used in the pitch contour and Teager Energy methods presented below. The Hirsch and Wigele scoring method uses the raw output of the cepstrum.

Speech production is modeled as an impulse train driving a linear filter. The frequency of the impulse train determines the pitch of the voice. The transform of the speech signal according to this model is thus $Y(s) = X(s)H(s)$, where $H(s)$ is the transfer function modeling the mouth and throat, and X(s) is an impulse train that models the voice box.

The cepstrum of a signal is defined as

$$C(x) = IFFT(\log_{10}(abs(FFT(y[n])))$$

The logarithm above converts the multiplication above (in frequency) to addition (in "quefrency"). Therefore, the cepstrum is composed of a low-quefrency component that reflects the state of the mouth and throat plus a high-quefrency component that reflects the pitch of the voice. In fact, the maximum of the cepstrum occurs at the fundamental quefrency of the voiced segment. This is converted back into frequency with the simple relation

$$F_0 = F_s/i_{max}$$

where i_{max} is the index corresponding to the maximum of the cepstrum.

The cepstrum is, of course, calculated on windows. For this project, 512 point Hamming windows were used. To avoid complications related to noise, the search for maxima is restricted to the range 40 Hz to 240 Hz, where most voices lie. Even with this defense, this pitch detection algorithm suffers from spiking and pitch doubling (an artifact of the cepstrum); median filtering (which we employed) takes care of these problems quite well.

III. Detection Methods

Pitch Contour

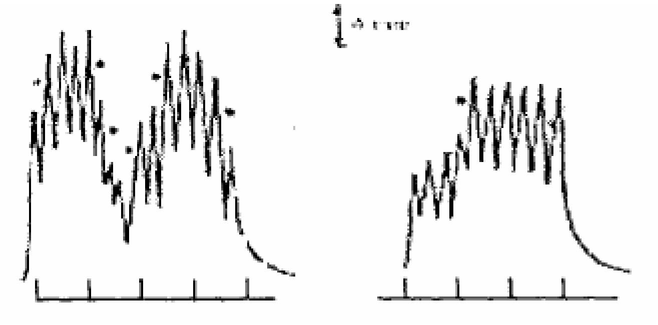
The pitch contour method for lie detection is a relatively simple algorithm. It is based on the premise that each person's voice has a unique fundamental frequency (F0), which remains fairly constant when the person is not under stress [2]. If the person is under stress (which should be the case when the person is lying), the fundamental frequency of their voice will fluctuate. Gadallah, et al. propose in [2] that a variation greater than 5 Hz indicates that a person is under stress.

This algorithm was implemented in several steps. First, the pitches of several training utterances are found through cepstral analysis. These pitches are averaged and used as the baseline value. The fundamental frequency is then found for each subsequent utterance spoken and compared to the baseline value ($\Delta F = |F0 - avg(F0)|$). If the difference is greater than or equal to 5 Hz, the utterance is determined to be a lie. If the difference is less than 5 Hz, the person is unstressed and presumably telling the truth.

As a stress detector, pitch contour boasts an accuracy rate of greater than 90% [2]. It is important to note though that the authors of [2] intended the method to be used for stress detection. The method was not suggested for lie detection. However, the theory behind the physiological manifestations of lies supports our extension of this algorithm from stress detection to lie detection.

Hirsch and Wiegele Scoring

The Hirsch and Wiegele (H & W) scoring method is an algorithm that tries to improve upon the poor reliability of the Psychological Stress Evaluator (PSE). H & W scoring is based on the premise that the amplitude of the fundamental frequency (F0) becomes less erratic when an individual is under stress [2].



Unstressed Utterance Stressed Utterance

Figure 1: Comparison of stressed and unstressed utterances (taken from [2])

For each utterance, the speech signal is broken up into windows, and the cepstrum is calculated for each window. The maximum amplitudes of adjacent windows are compared, and if the difference between consecutive amplitudes is greater than 10% of the maximum amplitude of that utterance, the score is increased by one; otherwise it remains constant. The total score is then divided by the number of frames in the utterance. This result is the Hirsch and Wiegele score for that utterance. The definition is given by

$$H\,\&W = 2/v \sum(i = 0,v)\ k(i)$$

$$\text{where } k(i) = 1 \text{ for } |A(i) - A(i+1)| > 0.1{*}A_{max},\ 0 \text{ otherwise}$$

A(i) is the maximum amplitude of the cepstrum of the ith frame, and

v = number of voiced frames

A baseline is established as the average H&W score over several unstressed training utterances. Utterances under test are then classified as stressed if there is a significant change in the H&W score from the baseline. The threshold was empirically determined to be 0.12.

As a stress detector, the Hirsch and Wiegele scoring method was reported to have an accuracy rate of greater than 70% [2]. Yet, it is important to note that the reference for this method intended the method to be used for determining when a person is/is not under stress. The method was not suggested for lie detection. However, the theory behind the physiological manifestations of lies supports our extension of this algorithm from stress detection to lie detection.

Teager Energy Operator

According to traditional linear acoustic theory, speech production is a linear operation. However, recent studies have shown that in fact speech production is a non-linear process, which can be decomposed into AM and FM components [3]. If an individual is under stress, it is expected that the pitch of his voice will be more variable, and the FM component will show more fluctuation. Thus the Teager Energy Operator (TEO), which was originally developed in the context of fluid dynamics, can be used to estimate the energy of the airflow within the vocal tract and extract the AM and FM components of the speech signal. The TEO is defined for discrete-time signals as

$$\varphi[x(n)] = x\hat{}2(n) - x(n+1)*x(n-1)$$

where x(n) is the sampled speech signal. If we assume the signal is a single FM-modulated signal, that is r(n) = a(n)cos(2*pi*f(n)), the FM component of the signal can be approximated as

$$f(n) = 1/(2*PI*T_s) * \arccos(1 - ((\varphi[y(n)] + \varphi[y(n+1)])/4*\varphi[x(n)]))$$

where y(n) = x(n)-x(n-1), the time-domain difference signal.

In practice, speech consists of several FM modulated signals: one at the fundamental frequency and several around the various formants. Thus, in order to use the TEO to extract the FM component of the speech signal, we must first filter around the fundamental frequency; this is dependent on the pitch detection algorithm presented earlier. A Butterworth passband filter centered at the fundamental frequency, with a 3 dB bandwidth of F0/2 and 10 dB of attenuation in the stopband produced acceptable results. In addition, the TEO misbehaves during unvoiced segments, so accurate voiced/unvoiced detection is crucial to any stress detection scheme based on the TEO.

Shown below, in Figure 2, are graphs comparing the FM components of two utterances. The figure on the left is the FM component of a truth. The figure on the right shows that of a lie. As is readily apparent, these two are significantly different. The pitch of the truth oscillates fairly regularly around 100 Hz, whereas the lie is vastly more variable. Most of the samples from our polygraph database followed this trend, however specific numbers are not available.

(A considerable amount of massaging was required to generate the plots below. This stemmed from an incomplete knowledge of the TEO's behavior, especially at the boundary between voiced and unvoiced segments. In addition, the TEO has a tendency to spike and throw off variance calculations. In the end, we decided not to implement this on the EVM, because time constraints did not allow us to develop a robust detection algorithm based on the TEO. These plots are provided to demonstrate the potential the TEO offered, which we were unable to fully exploit.)

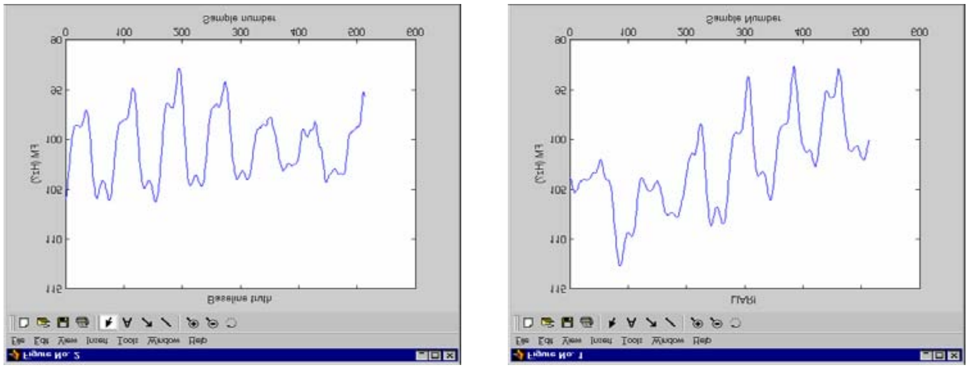


Figure 2: Comparison of FM components of a truth and a lie

IV. Databases

In order to test the methods presented above, a specialized database of three polygraph sessions was assembled. Mr. Morris Ragus, a certified polygrapher, administered a standard Pre-Evaluation computerized polygraph test to two group members. This test is usually administered before an actual polygraph test, in order to gauge a subject's suitability for testing. A transcript of the polygraph sessions can be found in Appendix A. The questions and answers were recorded on a DAT using professional grade equipment, converted to .wav files, forming the data sets “Eric,” “Andrew1,” and “Andrew2.” It should be noted that although only two lies were expected for each data set, "Andrew1" indicated deception on three questions. Thus a second administration was required to clear up the doubt.

The drawbacks of using this database are that it is unfortunately small. There were only two participants, both male, and very few questions under test. However, it was the only database of lies that was available, and showed that lie detection using voice is possible. The authors sincerely thank Mr. Ragus for the time he volunteered in providing background information and administering the polygraph exams.

In addition to the home-brew database of lies, the Speech Under Simulated and Actual Stress (SUSAS) database was also used for testing. SUSAS was created by John H. L. Hansen through the Linguistic Data Consortium at the University of Colorado Boulder. The SUSAS database contains hundreds of recordings of spoken words, with all of the utterances coming from a 35-word vocabulary set. The recordings are divided into four separate categories: Talking Styles Domain (simulated stress – spoken slow, fast, angry, question, soft, loud, clear), Single Tracking Task Domain (calibrated workload tracking task – moderate and high stress, Lombard effect), Dual Tracking Task Domain (acquisition and compensatory tracking task – moderate and high stress), and Actual Speech Under Stress Domain (amusement park roller-coaster, helicopter cockpit recordings – g-force, Lombard effect, noise, fear, anxiety). For each stressed utterance recorded, a complimentary unstressed utterance was spoken and recorded. There are 32 speakers, both male and female, ranging from 22 years to 76 years of age. Accents ranging from Boston to New York to "general." The data was sampled at 8000 Hz and 16-bits per sample.

This database however is unsuited to the application of lie detection. In most of the samples, there are audible changes in pitch (some as large as pitch doubling) which are not heard during lies. However, it is a useful check on the methods presented in this paper and could have been used had the previous database not been assembled.

V. Implementation

Preexisting Code

There exists a significant amount of public domain speech processing code. One such library is the Speech Processing Toolkit [4], used by Group 1 last year. This library contained code for calculating the cepstrum and fundamental frequency of a voiced utterance. However, the library did not compile due to a broken makefile. The code did provide a useful example, which we followed in the code that was eventually ported to the EVM.

In addition, TI provides standard radix-2 FFT/IFFT code for the EVM [5]. However, much of it did not compile, and the code that did compile did not work with interrupts. Bizarrely, the "equivalent C code" provided in the comments of the assembly listings compiled and worked flawlessly, and it was those functions that were ultimately used.

Finally, Lab 1 provided a skeleton for communication between the CODEC and the EVM, and Lab 2 contained code for initiating PCI transfers between the PC and EVM. The authors wrote all of the remaining code – voiced/unvoiced detection, cepstral analysis, pitch detection, Hirsch and Wiegele scoring, Teager Energy Operator, automatic activation when input offered to the CODEC, and code for reading .wav files.

Version 1 – Real-time

The first implementation of the lie detector ambitiously attempted to detect lies in real-time from speech sampled by a microphone. The EVM read the CODEC, and after a frame of 512 samples was received, the 16-bit signed integer samples were converted to floating point numbers and normalized to -1.0 - 1.0. It is on these floating point numbers the EVM calculated the energy of the current frame. If the energy fell below 10, the frame was considered background noise and no further processing was performed.

If the frame was considered speech, the frame was multiplied by a 512 point Hamming window (which was hardcoded into memory to save some cycles) and the cepstrum was calculated. The energy, pitch, and cepstral maximum were sent over the PCI bus to the PC. Once an entire utterance was received, the PC would classify the individual frames as voiced or unvoiced. It then calculated the median of the pitches of the voiced frames and called that the pitch of the utterance. This pitch is compared to the baseline according to the pitch contour method, and a decision about the presence of stress was rendered accordingly. The PC also calculated the H&W score for the utterance, and classified it according to the algorithm outlined above.

Unfortunately, there were serious issues surrounding this real-time implementation. The problematic TI code had issues working with interrupts, and thus the cepstrum calculations were suspect. In addition, depending on the position in the internal buffers that the speech started, the calculated pitches varied wildly and were generally unreliable. Thus, this method had to be abandoned for testing the algorithms.

Version Two - Offline

The offline method that was eventually implemented is basically the same as the previously mentioned real-time method; the EVM calculates the cepstrum and energy of each frame, and the PC interprets the results and classifies the speech accordingly. However, instead of reading the speech samples from the CODEC, the EVM receives the speech samples from the PC, which in turn parses standard .WAV files. This method proved much more reliable, since the speech always started in the same position in internal buffers and the voiced/unvoiced detection gave consistently correct results.

Memory and Speed

Memory Usage

- 16-bit mono window: 512 * 2 bytes = 1 KB
- 32-bit floating point window: 512*4 bytes = 2KB
- Cepstrum output: 512 * 2 * 4 bytes = 4 KB
- Hamming window: 512 * 4 bytes = 2 KB
- Twiddle factor table: 512 * 4 bytes = 2 KB

Very little memory (comparatively) is need for this application. All necessary buffers fit comfortably in on-chip memory.

Speed

The bulk of the processing involves calculation of the cepstrum of each window. Therefore, the following numbers should be considered representative of the total execution time.

Average cycles spent in cepstral_analysis() per call:

- No optimization: 1440387 cycles
- Level 0 optimization: 671503 cycles
- Level 1 optimization: 406633 cycles
- Level 3 optimization: 233636 cycles

There would be no problem running this code in real-time.

Results and Discussion

Results

The pitch contour method, as presented above, performed poorly at detecting lies. Reliability was around 50%, which is useless. However, it detected stress in the SUSAS database reasonably well.

Hirsch & Wigele scoring fared much better at detecting lies. As can be seen in Table 1, it was 67% accurate (or 75%, depending on whether fooling a polygrapher counts as a lie or not) on the "Andrew1" database, and a remarkable 100% accurate on "Andrew2." However, it performed quite poorly on "Eric1." The most reasonable explanation for this is that the subject was tired at the time of examination, and this tiredness may have masked any changes in his voice.

ANDREW1	ANDREW2	ERIC
Truth	Truth	Truth
Lie	Lie	Truth
Lie (X)	Truth	Truth (X)
Lie	Truth	Lie (X)
Lie (X)	Truth	Lie (X)
Lie	Lie	Lie (X)
		Lie
67% (75%) Accuracy	100% Accuracy	42% Accuracy

Table 1: Results of H&W test on polygraph database

Future Work

The Teager Energy Operator is fairly unexplored in its potential for stress detection, and more specifically lie detection. There are several other algorithms for stress detection that use it as its basis that are outlined in [3]. During the course of this project, a MATLAB implementation of the algorithm presented in this paper was designed. In fact, even a C version was developed, but never ported to the EVM. However, as demonstrated by Figure 2, it could be a powerful tool in extracting features that indicate stress and, by extension, deceit.

References

[1] S. Ahmadi and A. S. Spanias, “Cepstrum-based Pitch Detection Using a New Statistical V/UV Classification Algorithm,” *IEEE Trans. on Speech and Audio Processing*, vol. 7, no. 3, pp. 333-338, May 1999.

[2] M. E. Gadallah, M. A. Matar, and A. F. Algezawi, “Speech Based Automatic Lie Detection,” *Sixteenth National Radio Science Conference*, Ain Shams University: Cairo, Egypt, Feb. 23-25, 1999, vol. C33, pp. 1-8.

[3] G. Zhou, J. H. L. Hansen, and J. F. Kaiser, “Nonlinear Feature Based Classification of Speech Under Stress,” *IEEE Trans. Speech Audio Processing*, vol. 9, no. 3, pp. 201- 216, Mar. 2001.

[4] Speech Processing Toolkit, http://kt-lab.ics.nitech.ac.jp/~tokuda/SPTK/index-j.html

[5] Texas Instruments, http://www.ti.com/sc/docs/products/dsp/c6000/62bench.htm

Appendix A - Transcripts of Three Polygraph Sessions

Background:
Two participants from the group were interviewed. The first, Eric, chose one color out of a list of colors to be his deliberate deception color. When asked if he had picked any particular color, he was instructed to answer "NO," regardless of whether or not he had in fact picked that color. The second subject, Andrew, was tested in the same manner using numbers instead of colors.

Transcript: (Note: Each participant answered “NO” to every question.)

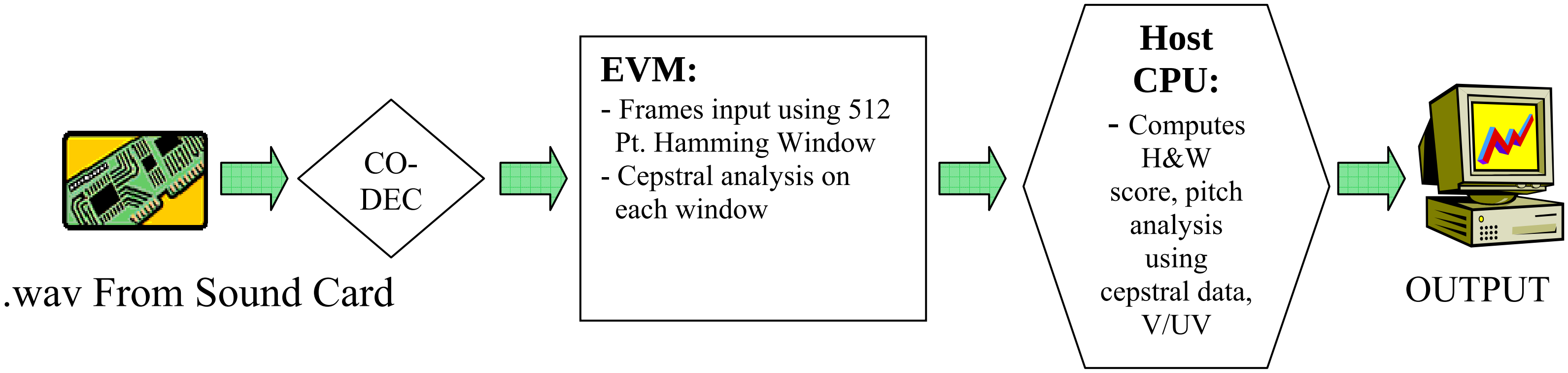
Polygraph Session 1 – Eric Carr
"Is your last name Carr?" (mistake, disregarded)
"Is your first name Carr?" (training/baseline answer)
"Did you print the color Green in the two circles?" (training/baseline answer)
"Did you print the color Black in the two circles?" (training/baseline answer)
"Did you print the color Brown in the two circles?"
"Did you print the color Purple in the two circles?"
"Did you print the color Grey in the two circles?” (deception color)
"Did you print the color White in the two circles?"
"Did you print the color Red in the two circles?"
"Did you print the color Blue in the two circles?"
"Did you lie to me about printing the color in the two circles?” (deception answer)

Polygraph Session 2 – Andrew Pomerance
"Is your last name Andrew?” (training/baseline answer)
"Did you write the number 0 in the two circles?” (training/baseline answer)
"Did you write the number 1 in the two circles?” (training/baseline answer)
"Did you write the number 2 in the two circles?"
"Did you write the number 3 in the two circles?” (indicated possible deception)
"Did you write the number 4 in the two circles?"
"Did you write the number 5 in the two circles?” (deception number)
"Did you write the number 6 in the two circles?"
"Did you lie to me about writing the number in the two circles?” (deception answer)

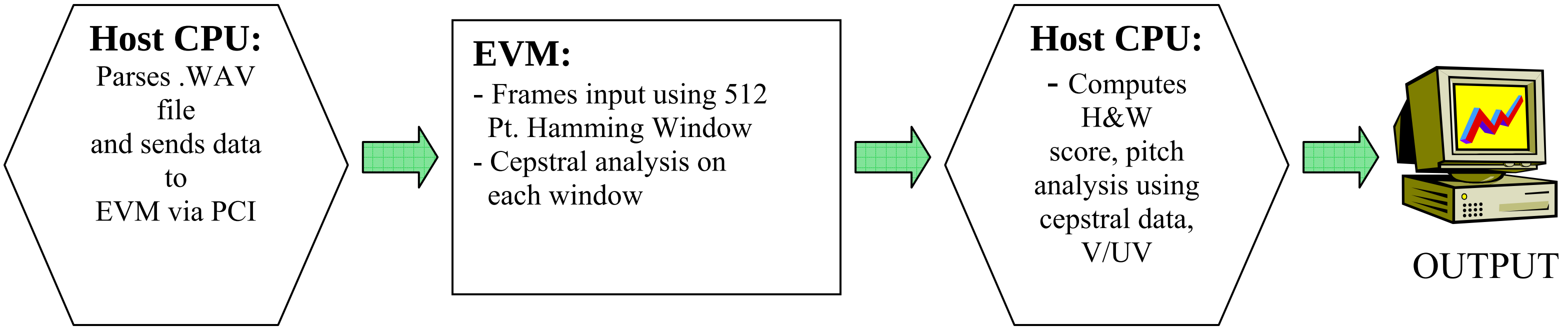
Polygraph Session 3 – Andrew Pomerance
“Is your last name Andrew?” (training/baseline answer)
"Did you write the number 0 in the two circles?” (training/baseline answer)
"Did you write the number 1 in the two circles?” (training/baseline answer)
"Did you write the number 6 in the two circles?"
"Did you write the number 5 in the two circles?” (deception number)
"Did you write the number 4 in the two circles?"
"Did you write the number 3 in the two circles?"
"Did you write the number 2 in the two circles?"
"Did you lie to me about writing the number in the two circles?” (deception answer)

Appendix B - Signal Flowcharts

Signal Flowchart 1


.wav From Sound Card
CO-DEC
EVM:
- Frames input using 512 Pt. Hamming Window
- Cepstral analysis on each window
Host CPU:
- Computes H&W score, pitch analysis using cepstral data, V/UV
OUTPUT

Signal Flowchart 2


Host CPU:
Parses .WAV file and sends data to EVM via PCI
EVM:
- Frames input using 512 Pt. Hamming Window
- Cepstral analysis on each window
Host CPU:
- Computes H&W score, pitch analysis using cepstral data, V/UV
OUTPUT