Acoustic Echo Canceling:
Echo Equality Index

Mengran Du, University of Maryalnd
Dr. Bogdan Kosanovic, Texas Instruments

Industry Sponsored Projects In Research and Engineering (INSPIRE)
Maryland Engineering Research Internship Teams (MERIT) 2007

Abstract:

Evaluating the performance of Acoustic Echo Canceling (AEC) systems in telephony for full-duplex hands free operation is a challenging digital signal processing problem. This project used Fuzzy Logic to design an intelligent fuzzy inference system (FIS) that assigned quality index to AEC based on its debug statistics during phone conversations. Variations on conversation environment such as single or double talk, background noise, volume, and Non-Linear Processing were tested to examine their effects on AEC EQI. MATLAB functions were developed to evaluate FIS with AEC debug statistics as inputs. However, further research is required to verify some parameters of the debug statistics before FIS can function properly.

Table of Contents

1 Introduction

Acoustic echoes are caused by speech signals leave the speakers of a phone, bounce off walls and objects in the room, then return through the microphone on the phone. This causes the user at the far end of the conversation that originated the speech to hear oneself speaking. In order to eliminate this echo, Acoustic Echo Canceling (AEC) has been widely used in teleconferencing applications as well as in telephony for full duplex hands free operation. The Acoustic Echo Remover (AER), which embodies both the Acoustic Echo Canceller (AEC) and the Acoustic Echo Suppressor (AES), is a component of the VoIP phone that is designed to predict and remove far end acoustics echoes caused at the near end. For the purpose of this project, the terms AER and AEC will be used interchangeably.

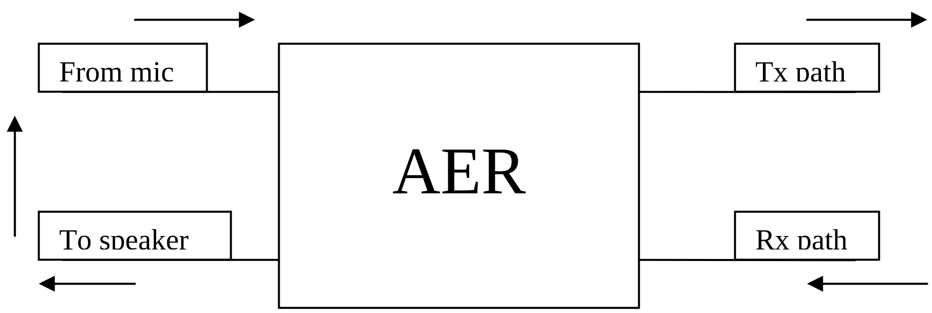

Figure 1: Echo path through near end AER

Although the AEC has already been implemented in many VoIP phones, it is still a challenging digital signal processing problem to evaluate the performance of these AEC systems due to various conditions, e.g. very strong signal power, nonlinear distortion, time-varying acoustic echo path. This project will follow the footstep of a previous project at Texas Instruments, in which Line Echo Canceller (LEC) performance was studied. Line Echo Canceller is a part of the public switched telephone network (PSTN) that reduces echo caused by mismatched impedance at the 4-wire phone circuit to 2-wire circuit junction. A Fuzzy Inference System (FIS) was created to evaluate LEC performance. The FIS used fuzzy logic to define the performance of an LEC as a degree of on the scale of 0 to 1. This project was the first project on Acoustic Echo Canceling performance, and was carried out in a similar fashion as the LEC project.

Acoustic echo is most apparent when near end phone is operating in hands free or speaker mode, which is the common mode used in teleconferencing. During hands free mode, the volume of the received speech is much greater than handset mode. Therefore hands free echo power is significantly higher than the echo produced in handset mode. This project chose to focus on hands free operation, which would force AER to operate at 100%.

The first part of the project would consist of using the LEC FIS on AER debug statistics to get an overall comparison on LEC and AEC. Then the parameters for the inputs of LEC EQI would be adjusted to suit the AEC. The resulting Fuzzy Inference System of this project will be used in the future to assist in optimizing performances of IP telephony network and detecting problems in configurations of the IP Phones.

1.1 Acronyms

Throughout this document, several important terms and parameters are being used. A brief description of these parameters is presented here.

- **Near end** – the side of a telephony connection which contains the echo path on which the echo canceller is intended to operate. The echo canceller at this end is being tested.
- **Far end** – the side of a telephony connection that acts as a dummy device, which sends signals to the near end IP phone.
- **Rx path** – receive signal path (from network to the IP phone speaker) – the associated signal is also known as the far end signal
- **Tx path** – transmit signal path (from IP phone microphone to the network) – the associated signal is also known as the near end signal
- **Single talk** – the condition of having considerable activity only in the Tx path
- **Double talk** – the condition of having considerable activity in both the Rx and Tx path.
- **Echo Return Loss (ERL)** – the attenuation of a signal from the speaker to the microphone on the near end phone.
- **Echo Return Loss Enhancement (ELRE)** – for the purpose of this project, it refers only to the attenuation of the echo signal as it passes through the adaptive filter in the Tx path. The adaptive filter predicts the amount of echo based on Rx signal strength and background noise level, and then subtracts the estimated echo from Tx signal.
- **Combined Loss (Acom)** – for the purpose of this project, it refers only to the sum of ERL and ERLE.
- **Non-Linear Processor (NLP)** – a part of the AER that provide further cancellation in the Rx and Tx directions after adaptive filtering.
- **Fuzzy Inference System (FIS)** – a system that uses fuzzy reasoning to map an input space to an output space.
- **Hands free operation** – using speaker on the near end phone instead of handset.

2 Methods and Materials

2.1 Test scenarios

The experiment used a combination of 4 variables to create all the difference test scenarios possible. This way, the effect of each variable could then be examined individually. Also, two different speech signals and two different background noises were tested with each test scenario to obtain a more generalize AER performance. The test variables included:

- Single talk hands free or double talk hands free operation
- No Tx noise, moderate Tx noise (-50dBm), high Tx noise (-30dBm)
- NLP on or off
- Nominal volume or maximum volume
- Different test signal and different noise combination

2.2 Materials and Hardware Setup

The materials used in this project included:

- PC with MATLAB, Adobe Audition 2.0, TeraTerm, and Ethereal installed

- 2 x 11.20 SVCA IP phones with operating system loaded
- Telephone Handset Audio Tap (THAT-2) box
- Ethernet cables
- Ethernet to serial adapters
- Serial to 3.5mm stereo jacks
- 3.5mm stereo splitters to left and right channel RCA
- Male-to-male RCA adapters
- Female-to-female RCA adapters
- BNC Coaxial cables
- PC speakers with amplification

The near end IP phone was placed in the middle of the table in the quiet room. A set of PC speakers was placed 40cm away from the microphone of the IP phone. The far end IP phone was placed next to the computer station outside the quiet room. The far end IP phone was connect to a THAT box, which was then connected to the left audio channel on the computer. The THAT box was used so that a speech files can be played from the computer through the far end IP phone to the near end IP phone. This simulated the far end speech. The right audio channel of the computer was connected to the speakers inside the quiet room. Playing speeches on the right channel of the computer to the speakers simulated near end speech and near end (Tx) noise. Both IP phones and the computer were connected to a hub via Ethernet. The computer was also used to telnet to the near end phone to request debug statistics. The quiet room was closed and cleared of any personnel. All of the test scenario adjustments and volume configurations were done remotely on the computer.

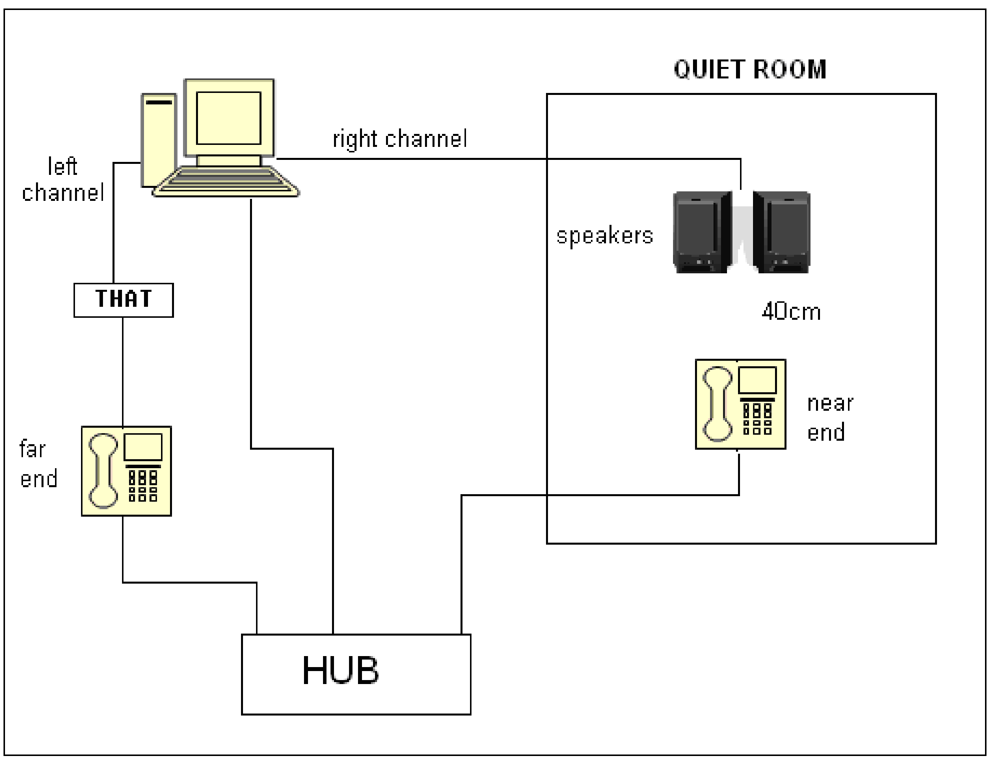

Figure 2: hardware setup

2.3 Data collection

The language Expect is a derivation from the language Tcl. It is used in general to automate commands in environments such as telnet. For this project, an Expect script was written to automatically telnet to the near end IP phone and send commands in a periodic fashion to request for debug statistics. AER debug statistics was requested every 2 seconds for 70-80 seconds depending on the signal. Each set of debug statistics sent back was a 60 entry hexadecimal vector. The entire sequence of commands and hexadecimal matrices were saved to a text file, which was later parsed using MATLAB to acquire the actual decimal numbers.

3 Debug Statistics Analysis

The 4 variables, 2 signals and 2 noises combined to a total of 96 test scenarios. The raw debug statistics for each test was saved to a text file. A MATLAB to extract and converted all the hexadecimal numbers to decimal format with the appropriate units of dB, dBm and seconds. In addition to the measurements contained in debug statistics, other performance parameters for the AER needed by the LEC FIS were also calculated. The resulting output by this MATLAB function was a matrix containing collected debug statistics, calculated measurement, arranging by the time of the debug statistics request.

Acom calculations

3 different Acom were included in the matrix, which used different measurements of ERLE. For the purpose of this project, Acom was only the sum of ERL and ERLE, which represented only physical signal attenuation as it passes through air and attenuation caused by adaptive filter.

- Acom = ERL + maxERLE
- Acom = ERL + currERLE
- Acom = ERL + avgERLE

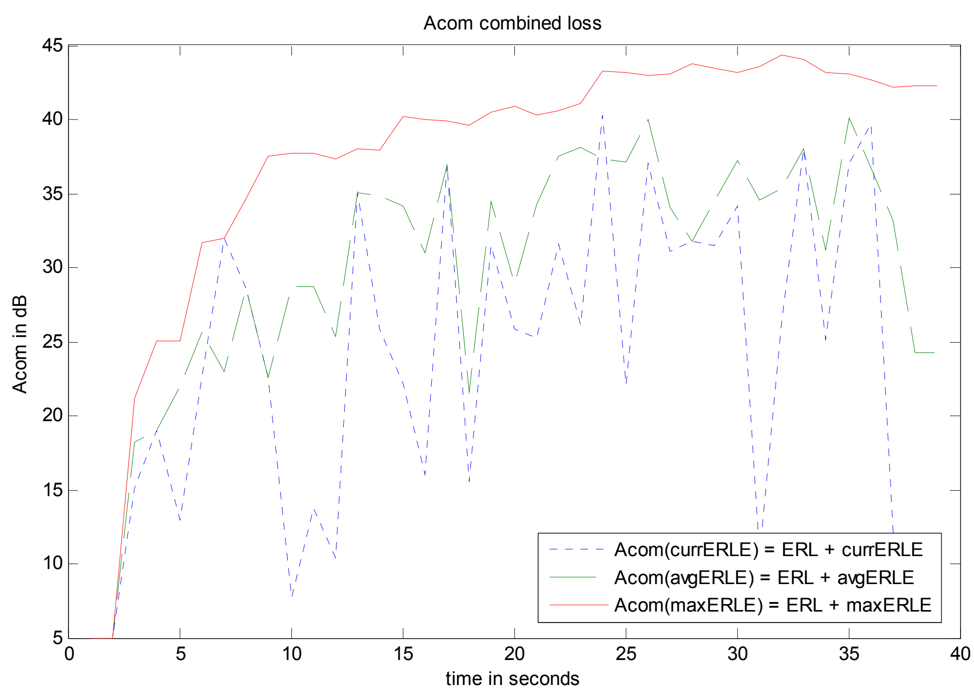

Fig 3: three different Acom measurements

The Acom(currERLE) calculated using current/instantaneous ERLE had many short-term fluctuations, which would cause EQI fluctuation. This was because of Acom's dominance in EQI calculation (refer to AER Fuzzy Inference System section). The exponentially averaged ERLE was used to obtain Acom(avgERLE). This Acom(avgERLE) was found to be a better representation of the signal, because it had less fluctuation and matched closest with captured Acom levels measurements by Ethereal packet sniffer. The third Acom that used maxERLE represented the best Acom recorded in the past. Therefore its shape was flatter and saturated quickly. Because both Acom(avgERLE) and Acom(maxERLE) could be used to represent the long time behavior of the AER, they were both chosen for EQI calculated, as inputs to the FIS.

Average Tx and Rx speech power calculations

In the time domain, speech powers also had many short term fluctuations. This project studied the performance of the AER in a long period of time, so exponentially running averaging was used in a similar fashion as Acom to decrease fluctuation. The resulting averaged Tx and Rx signal powers were calculated using exponential moving average parameters Tau = 4 and alpha = period/Tau = 2/4=.5. The Tau represented the number of samples used in averaging and the smoothing factor alpha represented the degree of weighting. The averaged Tx and Rx signal powers had their initial values set to -20dBm until the first Tx or Rx speech activity was detected, and the averaging process was then started.

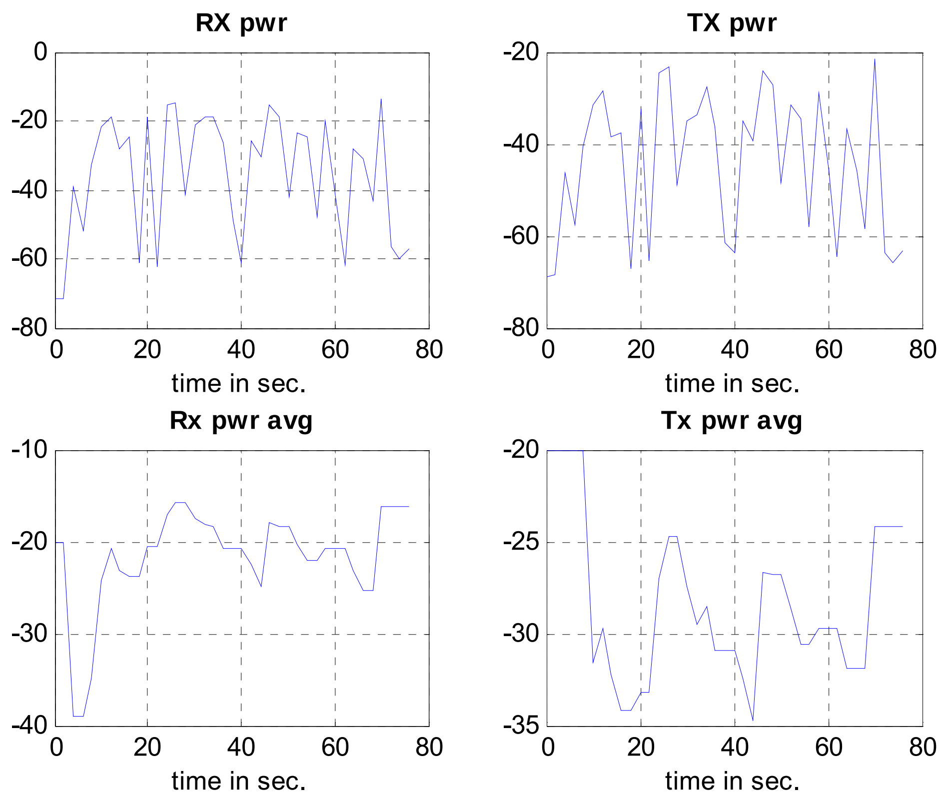

Fig 4: original speech power vs. exponentially averaged speech power

4 EQI Analysis

4.1 Evaluating LEC FIS with AER debug statistics

After testing all 4 sets of signal/noise combination, all of the necessary FIS inputs were obtained from the debug statistics. The second MATLAB function was written to select the five input variables ERL, Acom, Tx noise, Tx/Rx ratio, Rx power from the debug statistics matrix, and then inputted these variables into LEC FIS, outputting an EQI value for the test.

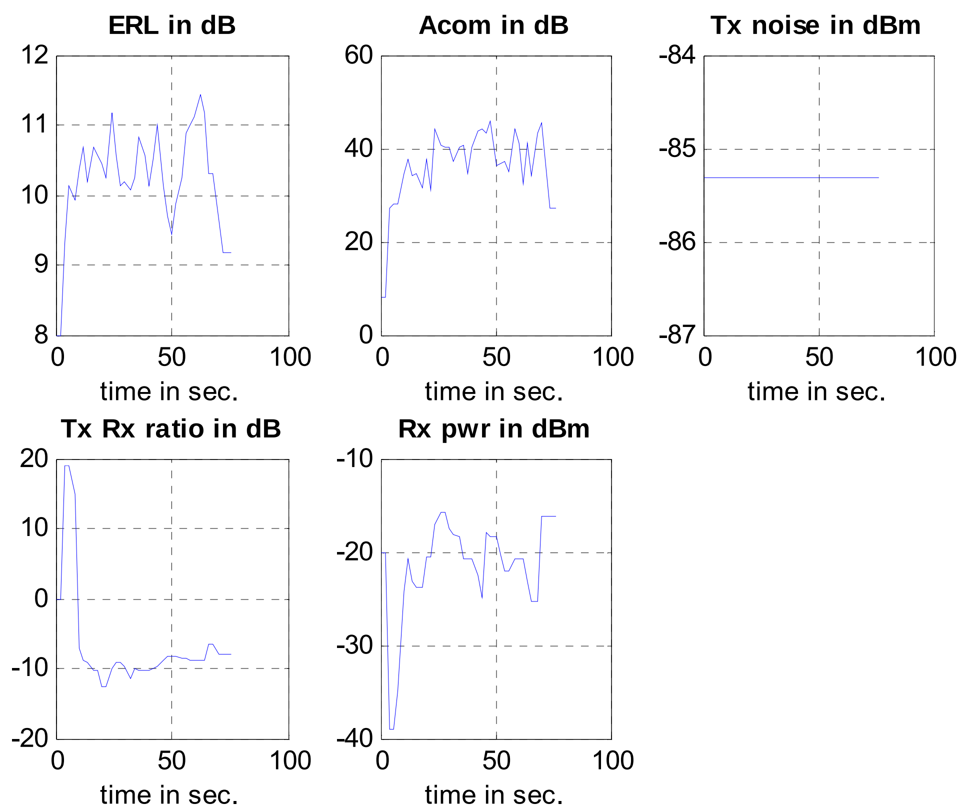

Fig 5: five FIS inputs

4.2 AER Fuzzy Inference System

The LEC FIS was first used with AER debug statistics to obtain EQI values. This FIS obtains a fuzzy value between 0 and 1 for the echo canceller performance. It takes in as input the ERL, Tx Noise, Acom, Tx/Rx ratio, and Rx Speech power.

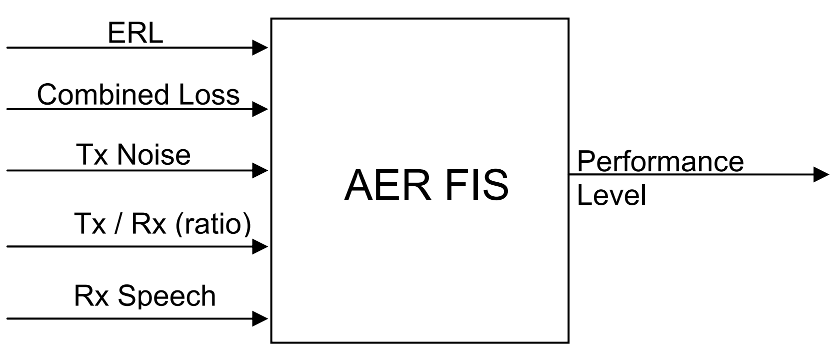

Fig 6: Fuzzy Inference System input vs. output relationship

Within the FIS, each input was defined as a membership function with specified limits and ranges. For example, the graphic representation of the input ERL is the following

ERL in dB

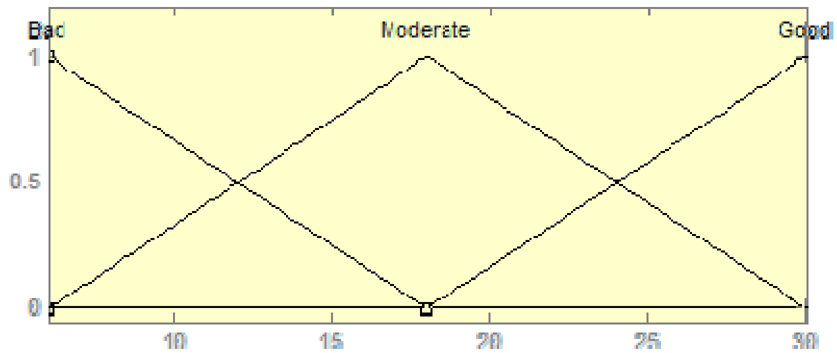

Fig 7: graphical representation of FIS membership function

Table 1: FIS input membership functions

Input Variable	Value	Fuzzy Interpretation
ERL	0 – 18 (dB)	Bad
	0 – 30 (dB)	Moderate
	18 – 30 (dB)	Good

The other four inputs Acom, Tx noise, Tx/Rx ratio, and Rx power were defined in a similar fashion as the ERL. However, each variable had different parameters and different shapes, e.g. triangular, trapezoidal.

The performance output of the EC FIS was also defined by membership functions which specify the output ranges.

EC Performance

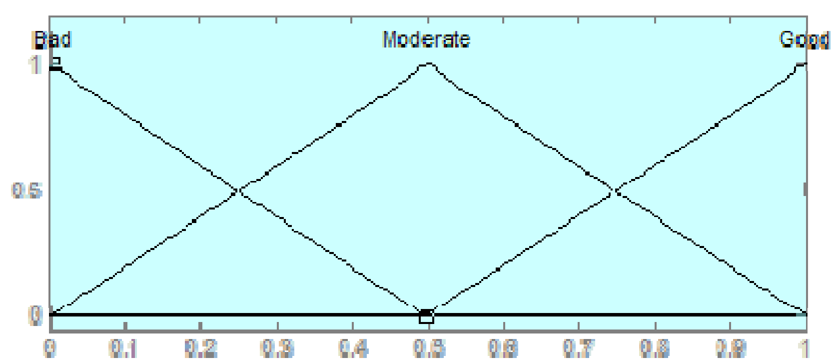

Fig 8: graphical representation of FIS membership function

Table 2: FIS output membership function

Output Variable	Value	Fuzzy Interpretation
EQI	0 – 0.5	Bad
	0 –1	Moderate
	0.5 – 1	Good

The EQI for each test case was evaluated by inputting the five inputs into the following fuzzy rules

1. If (Comb.Loss is Good) then (Performance is Good) (1)

2. If (Comb.Loss is Bad) then (Performance is Bad) (1)

3. If (ERL is Bad) and (Comb.Loss is Moderate) then (Performance is Bad) (1)

….

….

….

Fig 9: LEC FIS fuzzy rules

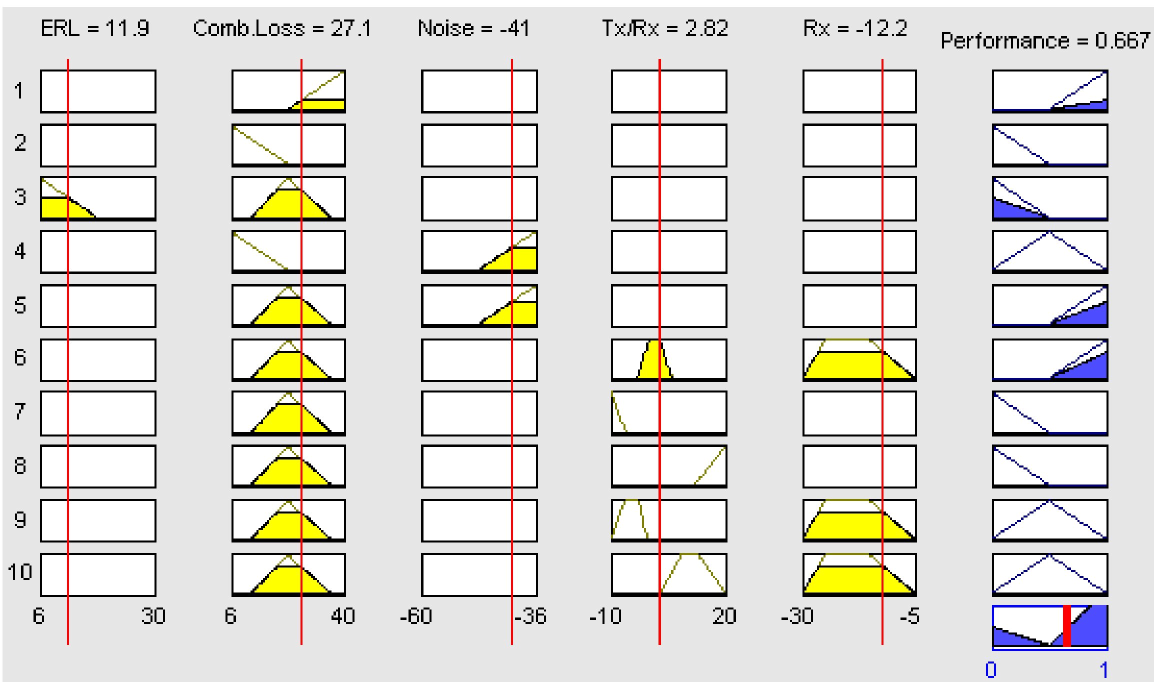

Fig 10: graphical representation of LEC FIS fuzzy rules

The rules that govern the functioning of the EC FIS were formulated to give importance to combined loss levels over the ERL and Tx Noise Levels. The performance of the Echo Canceller was, therefore, dominated by the combined loss level. However, if the ERL or Tx Noise for the signal is significantly lower than their desired value, the EC performance is affected negatively.

The weight assigned to each rule is indicated in **()** next to the corresponding rule. The maximum weight that can be assigned is unity (1). The rules for the EC FIS have been given an equal weight of unity.

EQI Averaging

The EQI graph resulted from the LEC FIS was a plot of instantaneous EQIs calculated at each request for debug statistics. This instantaneous EQI plot was averaged using the method of running sum. The averaged EQI at each time was the mean of previous EQIs. This averaging method was used to smooth out short-term fluctuations in instantaneous EQIs, and to obtain a mean EQI that could represent the average performance of the AER during that particular phone call. The mean EQI was the last value of the averaged EQI.

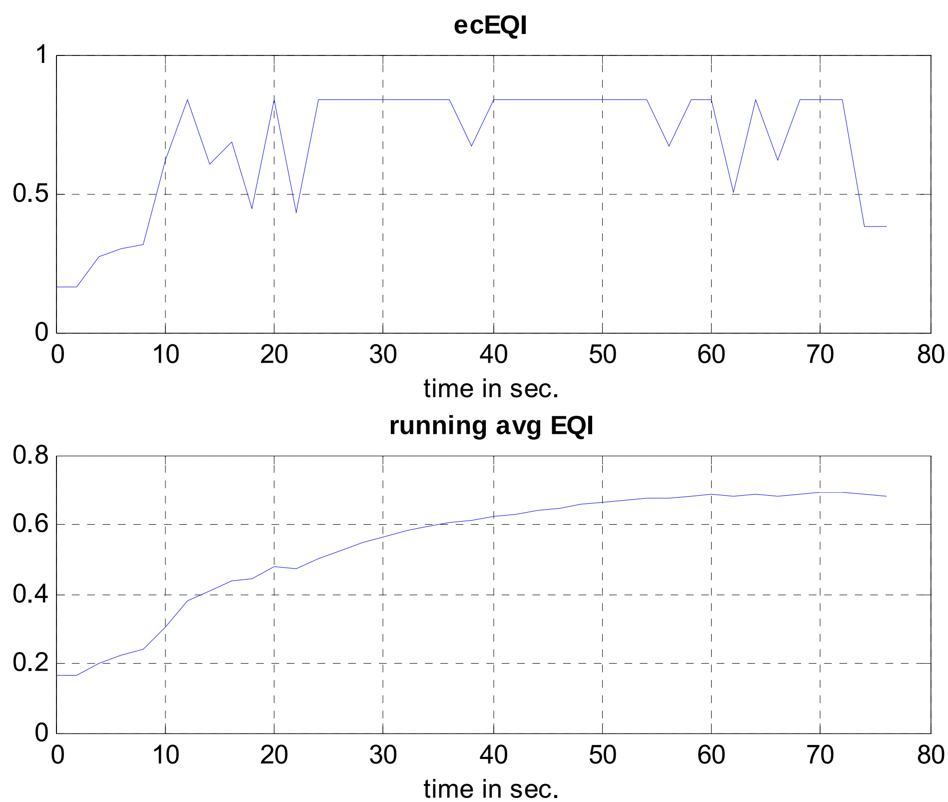

Fig 11: EQI and running averaged EQI

4.3 FIS Input Measurement Accuracy Verification

Before inputting the five inputs to the FIS, their measurement accuracy had to be tested. If these FIS inputs were measurement incorrectly by the AER DSP, then the resulting EQI would definitely be wrong. For each input, verification tests were designed to compare debug statistics with the raw captures of the speeches by Ethereal packet sniffer.

4.3.1 ERL

In order to sniff the pure Echo Return Loss, all of the AER and non-AER components in the Tx and Rx paths were turned off. At this time, white noise with RMS average of -10.43dBFS was played using Adobe Audition from the far end to the near end. The signal was allowed to exit through the speaker on the near end phone and return through the microphone as echo. Ethereal was used to capture the Rx and Tx signals. Adobe Audition was then used to measure the difference in power levels between the Rx and Tx signals. This difference was the accurate ERL, at 8.6dB.

To obtain the debug statistics, AER was then turned on, leaving all other components off. The same white noise was sent to the near end and returned as echo. The resulting debug statistics showed a saturation of ERL at 8.5dB. Repeated tests were taken to ensure the accuracy of ERL measurement.

4.3.2 Acom (ERLE)

Acom measurements were verified in a similar way as ERL. With all components except AER disabled, white noise was played from far end to near end. The Rx and Tx signals were sniffed, and their difference was calculated, which is now the accurate Acom measurement. Debug statistics were collected at the same time, and were compared with Adobe Audition measurements. While Audition Acom measurements increase from 15 -> 41dB over the course of the call, debug statistics showed an increase from 9.20 -> 44dB for Acom (maxERLE) and in crease from 9.28 -> 41dB for Acom (avgERLE).

These results showed that debug statistics Acom measurements were inaccurate at the beginning of the call, and it took some time to “train” itself to make an accurate measurement. The typical convergence time was 3-4 seconds. In the project, the tests conducted had durations of 70-80 seconds, so a 3-4 second convergence time was acceptable. Also, the Acom saturation values were within a 10% margin of the actual Acom, therefore making debug statistics Acom measurements acceptable to be used as FIS inputs.

The actual test signals were speeches spoken by human, however, had power levels fluctuations over the course of the call. Every time the speech fluctuated, the AER would take 3-4 seconds to make a correct Acom. This could have potentially brought some additional inaccuracy to the debug statistics Acom measurement.

4.3.3 Rx power, Tx power

During verification for Acom, the debug statistics collected also included Rx and Tx signal power measurements. These were compared with Ethereal captured Rx and Tx signals. The captured Rx signal had an average of -7.43dBm, and the debug statistics showed a range of -6 to -8 dBm. Therefore the Rx measurements for debug statistics were fairly accurate. Regarding Tx signal power, the captured Tx signal had an average of -14.03dBm, which was 8.6dB (ERL) lower than Rx signal. The debug statistics Tx signal powers were within the range of -14 to -16 dBm, which was approximate 8dB lower than the Rx signal powers, very close to the ERL measurements. These results indicate Rx and Tx signal power level measurements were very accurate, and thus acceptable to input to the FIS.

4.3.4 Tx Noise

Single Talk

For tests with no Tx noise, the AER debug statistics measured a power level of -85dBm to -85.5dBm. Because no Tx background noise was played and the phone was placed in the quiet room, this value of -85dBm represented near silence. For tests with moderate Tx noise, the -50dBm Tx background noise was measured as -85dBm as well. This was an incorrect Tx noise measurement. For tests with high Tx noise, the -30dBm Tx background noise was measured as -85dBm, slowly converging to a higher value. However, after 80 seconds of data collection, Tx noise for -30dBm case was still unable to saturate. Due to this slow convergence, Tx noise measurement for this case was concluded to be inaccurate as well.

No noise **-50dBm** **-30dBm**

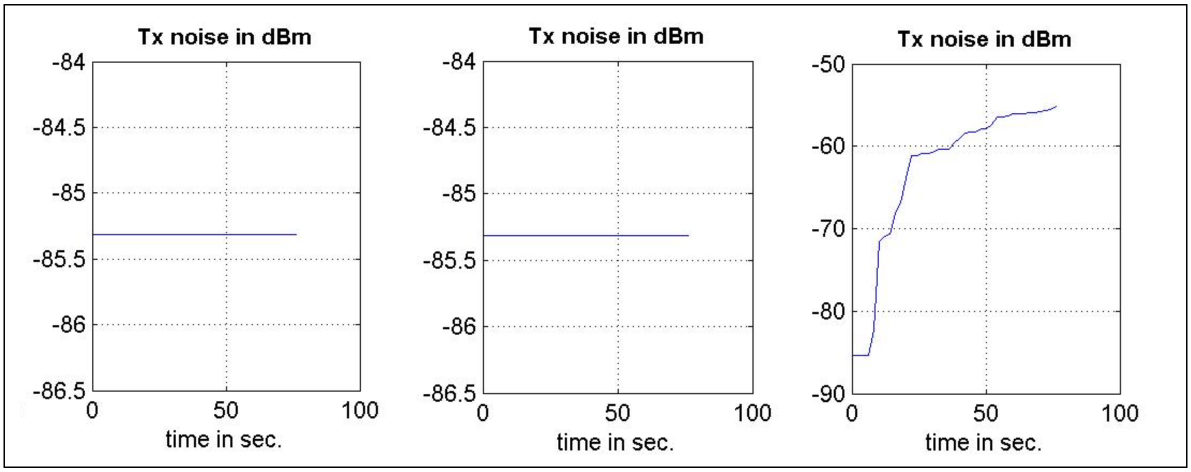

Fig 12: Tx measurement error in single talk mode

Double Talk

For tests with no noise, moderate noise, and high noise, the Tx noise measurements in AER debug statistics were identical. This suggested when only speech was played in near end background, some of the speech was picked up as noise. And when a mix of speech and noise was played, the noise measurements were still inaccurate.

No noise **-50dBm** **-30dBm**

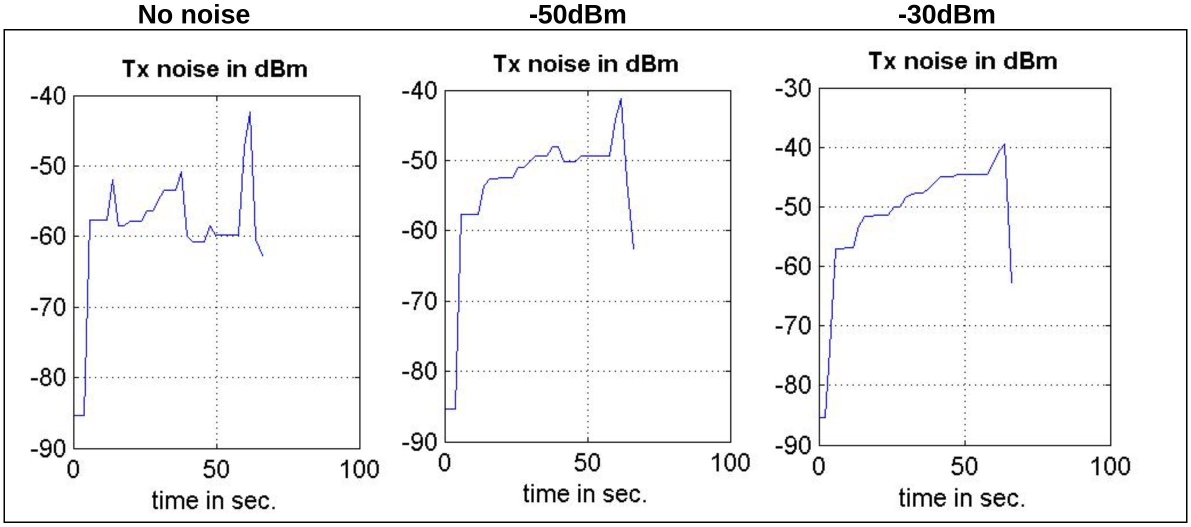

Fig 13: Tx measurement error in double talk mode

From these tests, the AER was determined to be unable to make the correct Tx noise measurement. In order to design a properly functioning AEC FIS, this Tx noise measurement needs to be corrected on the hardware level.

4.4 Variable Effects on EQI

Using the previously developed Line Echo Canceller Fuzzy Inference System, each acoustic echo test performed in this project was evaluated. Although Tx noise were determined to be inaccurate, its weight in EQI calculation defined by the FIS fuzzy rules was very small. This meant that the effect of Tx noise was relatively small, therefore this incorrect Tx noise input was still used. The change in EQI caused by each test variable was examined.

Table 3: Mean EQI for single talk mode

SINGLE TALK			Mean EQI calculated with Acom (avg ERLE)				Mean EQI calculated with Acom (max ERLE)			
	NLP	Volume	Sig1 babble noise	Sig1 office noise	Sig2 babble noise	Sig2 office noise	Sig1 babble noise	Sig1 office noise	Sig2 babble noise	Sig2 office noise
No noise	Off	Max	.5381	.6785	.4873	.4858	.7318	.7796	.6460	.6526
		Nom	.7564	.7775	.5834	.5665	.7877	.7853	.7110	.7008
	On	Max	.5617	.6484	.5037	.4988	.7473	.7641	.6108	.6444
		Nom	.7569	.7808	.6135	.6689	.7758	.7876	.7246	.7359
−50dBm noise	Off	Max	.4231	.5296	.3520	.4028	.7578	.7636	.5482	.6485
		Nom	.4276	.5750	.4618	.3279	.7606	.7841	.7362	.6636
	On	Max	.4233	.3699	.3829	.3003	.7206	.6804	.6274	.5234
		Nom	.4829	.5423	.5203	.4297	.7688	.7784	.7466	.7103
−30dBm noise	Off	Max	.1929	.1819	.1981	.1861	.2849	.2327	.3162	.2711
		Nom	.1681	.1633	.1751	.1725	.2103	.2010	.2099	.1954
	On	Max	.1821	.1721	.1932	.1721	.2145	.2347	.2379	.2413
		Nom	.1792	.1741	.1663	.1633	.2530	.2396	.1975	.1704

Table 4: Mean EQI for double talk mode

DOUBLE TALK			Mean EQI calculated with Acom (avg ERLE)				Mean EQI calculated with Acom (max ERLE)			
	NLP	Volume	Sig1 babble noise	Sig1 office noise	Sig2 babble noise	Sig2 office noise	Sig1 babble noise	Sig1 office noise	Sig2 babble noise	Sig2 office noise
No noise	Off	Max	.2237	.2174	.3704	.3382	.2974	.3492	.6373	.5067
		Nom	.2092	.2086	.3448	.3216	.3725	.3145	.6284	.6179
	On	Max	.2334	.2564	.3343	.3811	.3092	.3682	.5684	.5442
		Nom	.2018	.3096	.3642	.3202	.2539	.4259	.6984	.6118
−50dBm noise	Off	Max	.2143	.1949	.2731	.2826	.3085	.2725	.5463	.4911
		Nom	.2010	.2954	.2483	.2213	.3184	.4274	.4609	.5238
	On	Max	.2053	.2476	.2876	.3248	.3514	.3820	.5687	.5489
		Nom	.2510	.2551	.2866	.3007	.3363	.3579	.6063	.5956
−30dBm noise	Off	Max	.1838	.1926	.2530	.2365	.2502	.2684	.3566	.3426
		Nom	.1679	.1902	.2217	.2192	.1906	.2310	.2643	.2805
	On	Max	.1969	.2079	.2526	.2394	.2568	.2640	.2967	.2982
		Nom	.1761	.1802	.2492	.2252	.2384	.2481	.3464	.3516

4.4.1 Nominal volume vs. maximum volume

The volume mentioned here referred to the Rx amplification setting on the near end phone, commonly known as the speaker volume. The general trend in the EQI results showed that tests with nominal speaker volume produced a slightly higher EQI than tests with maximum speaker volume. This is due to the fact that the nominal volume tests had approximately 3dB higher Acom measurements than maximum volume tests. Since Acom is the most dominating input in EQI calculation, this 3dB difference in Acom caused nominal volume EQI to be slightly higher than maximum volume.

However, tests under high noise showed the opposite, in which maximum volume EQIs were higher than nominal volume EQIs. This was because when noise was high, Acom was lowered. This resulted inputs other Acom to dominate EQI calculation. For instance, under these conditions, Tx/Rx ratio and Rx power for maximum volume was higher than nominal volume. Defined in the fuzzy rules, this caused tests with maximum volume to have slightly higher EQI than tests with nominal volume.

4.4.2 NLP on vs. off

Non-Linear Processing had components in both the Rx and Tx direction, both of which caused attenuation. When calculating Acom in this project neither of the NLP components was considered. However, the Rx NLP still had indirect effects on the Acom. This was because measurements for ERL and Rx power used in Acom calculation were obtained after Rx NLP has performed attenuation on the Rx signal, thus NLP's on or off would have effect on the Acom and subsequently the EQI.

For tests with nominal volume, EQIs for tests with NLP on were slightly higher than tests with NLP off. But for tests with maximum volume, the EQIs did not show a correlation with NLP's status. This kind of observation could have been due to the fact that when speech levels were very high, the Rx NLP did not have much effect on the Rx signal. Rx NLP's effect was more apparent under nominal volume.

4.4.3 No noise vs. moderate noise vs. high noise

The EQIs clearly showed a negative correlation between the Tx noise level and EQI. Intuitively this made sense – as background noise increased, it became harder for the AER to find a reference power level to estimate the amount of echo, and thus the ERLE values decreased. This decrease in ERLE values directly impacted the combined loss Acom, and subsequently the EQI.

However, for tests with no noise or moderate noise at -50dBm, the EQI values calculated using Acom (max ERLE) were very similar. This was due to the parameters for the FIS input Acom. Because Acom (max ERLE) represented the maximum Acom level detected in the past, its value was usually much higher than Acom (avg ERLE). The allowed input parameters for Acom had a maximum of 40dB. Any Acom level above 40dB was mapped to 40dB. Therefore, even though tests with no noise had Acom (max ERLE) reaching 50dB, it appeared to the FIS same as test with moderate noise, which had Acom (max ERLE) approximately 40dB.

4.4.4 Single talk vs. double talk

In general, double talk tests had lower EQIs than single talk tests. The reason for this was similar to effects of noise levels. The AER used Tx background as a reference to estimate the amount of echo. Double talk tests, which were same as single talk tests with an additional speech played at near end, therefore behaved similar to single talk tests with loud Tx background noise.

4.4.5 Signal 1 vs. signal 2

In single talk tests, the EQIs for signal 1 were generally higher than signal 2. This was due to the nature of the two signals. The power levels for signal 1 were more constant than signal 2. This allowed easier echo cancellation on signal 1 than signal 2, which led to a slightly higher ERLE for signal 1. Also, signal 1 showed a higher ERL than signal 2 in most single talk tests. Together with the ERLE, signal 1's Acom measurements were higher than signal 2's Acom measurements.

However in double talk tests, the EQIs for signal 2 were higher than signal 1. The reason for this was that in double talk, signal 1's speeches were simultaneous, while signal 2's speeches were alternating. Looking at them as a conversation, signal 1 had 2 people speaking at the same time, while signal 2 was a more realistic conversation—one person spoke at a time while the other listened. Because of this, it was harder for AER to predict signal 1 echo levels than signal 2 echo levels. This led to a lower ERLE for signal 1 than signal 2, thus causing Acom and EQI to be lower as well.

5 Conclusion/Future Work

The EQIs obtained with LEC FIS reflected very well with the intuitive results for the test scenarios, therefore suggesting this FIS is functional for Acoustic Echo Canceller. Although Tx noise measurement was not accurate, its weight in EQI calculation was very small, so it only rendered a small error in the resulting EQI. If the Tx noise measurement was fixed on the hardware level, improvements can then be made on the AER. For future AER FIS improvements, ERL, Acom, Tx noise input parameters could all be modified to reflect better to AER debug statistics.

On top of the AER FIS, this project still produced some valuable MATLAB and Expect functions. For instance, the Expect script will be extremely helpful to signals engineers to collect debug statistics for AEC systems. The MATLAB functions automatically parsed and converted hexadecimal debug statistics into readable decimal matrix format. This would allow other engineers in the future to easily read, plot, and process AER performance data other than the five FIS inputs used in this project.

6 References

Acoustic Echo Canceller: User's Manual. Texas Instruments. May, 2007.

AEC Performance Statistics for PIQUA Statement of Work. Texas Instruments. May, 2005.

Kosko, Bart. Fuzzy Thinking: the New Science of Fuzzy Logic. New York: Hyperion, 1993.

Libes, Don. Exploring Expect. Beijing: O'Reilly, 1995

Network Simulation and Analysis Tool: User's Manual. Texas Instruments. 2005.