Low-Cost, High-Fidelity, Adaptive Cancellation of Periodic 60 Hz Noise[☆]

Kyle D. Wesson[*,a], Robert M. Ochshorn[b], Bruce R. Land[c]

[a] *Department of Electrical and Computer Engineering, Engineering Science Building, University of Texas at Austin, Austin, TX 78712*
[b] *Center for Advanced Visual Studies, Building N52-398, Massachusetts Institute of Technology, Cambridge, MA 02139*
[c] *School of Electrical and Computer Engineering, Phillips Hall, Cornell University College of Engineering, Ithaca, NY 14853*

Abstract

A common method to eliminate unwanted power line interference in neurobiology laboratories where sensitive electronic signals are measured is with a notch filter. However a fixed-frequency notch filter cannot remove all power line noise contamination since inherent frequency and phase variations exist in the contaminating signal. One way to overcome the limitations of a fixed-frequency notch filter is with adaptive noise cancellation. Adaptive noise cancellation is an active approach that uses feedback to create a signal that when summed with the contaminated signal destructively interferes with the noise component leaving only the desired signal. We have implemented an optimized least mean square adaptive noise cancellation algorithm on a low-cost 16 MHz, 8-bit microcontroller to adaptively cancel periodic 60 Hz noise. In our implementation, we achieve between 20 and 25 dB of cancellation of the 60 Hz noise component.

Key words: adaptive noise cancellation, embedded signal processing, noise cancellation on microcontroller, electrical interference

[☆] This work is a collaborative effort that took place from Spring 2008 to Fall 2009 at Cornell University.

[*] Corresponding Author

Email addresses: kdw24@cornell.edu (Kyle D. Wesson), rmo@mit.edu (Robert M. Ochshorn), brl4@cornell.edu (Bruce R. Land)

1. Introduction

60 Hz power line noise contamination can completely overwhelm signals of interest making sensitive or low-voltage measurements nearly impossible. This situation occurs in neurobiology laboratories when trying to make accurate measurements of low-voltage processes such as action potentials from nerve cells or when taking an electrocardiogram. It also affects other laboratory environments such as industrial instrumentation or chemical voltimetry which are high impedance measurements and, thus, noise sensitive.

A common way to eliminate the 60 Hz AC power line noise[1] from the signal of interest is with a fixed-frequency notch filter. The ideal notch filter has a flat frequency response over all frequencies of interest except at 60 Hz where the frequency response is zero. The ideal case cannot be achieved because any realizable notch filter will have two undesirable traits. First, the frequency response will not be flat, especially near the attenuated frequency where ringing occurs. Second, the notch frequency cannot vary and, thus, cannot attenuate noise with frequency shifts.[2] Studies have shown that the frequencies in 60 Hz power line references can vary as much as ± 1 Hz (Ramos, Mànuel-Làzaro, Rìo, and Olivar, 2007) making the notch filter unsuitable. Widening the notch to include 59–61 Hz will attenuate some of the desired signal and is thus impractical.

Adaptive noise cancellation systems can overcome the issues faced by the notch filter, and the underlying theory is well studied (Kuo and Morgan, 1996). In fact, much research has gone into digital notch filters with adaptive control over their variables (Ferdjallah and Barr, 1994). Typically, moving away from simple analog solutions leads to complicated implementation and a higher price tag. Although commercial systems such as "The Humbug" (Scientific, 2008) are available and adaptive noise cancellation systems have been implemented on DSPs or a FPGAs, these alternatives are expensive. Our method and implementation, using a low-cost microcontroller, is an excellent option for laboratory or classroom work on a budget because it can adapt to frequency variations in the noise and is a low-cost[3] alternative to

[1] 60 Hz power line noise is the focus of this paper although this method can be quickly and easily adapted in software to cancel 50 Hz line noise found in Europe.

[2] Actually, analog filters can vary due to aging and temperature shifts, but this is rarely desired behavior (Lynn, 1977).

[3] By using the phrase "low cost," we are not referring to a commercial production unit,

commercial systems. Our design is unique because unlike FPGA implementations (Ramos et al., 2007; Stefano, Scaglione, and Giaconia, 2005; Elhossini, Areibi, and Dony, 2006), our design maintains the signal in analog at full fidelity and passes the signal through the system with virtually no delay. And unlike direct sine-wave generation filtering techniques, as proposed by Ahlstrom and Tompkins (1985), our system can respond to the non-sinusoidal properties of powerline interference (Fig. 7 shows the complexity and higher harmonics of AC-line noise).

Our implementation uses an optimized least mean squares (LMS) adaptive filter (Kuo and Morgan, 1996). The goal is to periodically update a set of weights that modify the system output to minimize unwanted noise by eliminating the correlation between the error and reference signals. The output of the system (also known as the error signal) is fed back into the filter which modifies its weights according to the calculated correlation and optimized LMS algorithm. Sec 2.3 describes LMS algorithm modification and optimization required to implement it on a microcontroller.

2. Materials and Methods

2.1. High-Level Design

The goal of our cancellation system is to remove the correlation between the 60 Hz contaminated signal and the 60 Hz reference. The paper uses the following signal naming conventions:

- the *noisy* or *contaminated signal* is the signal to "clean" that has not gone through the system and contains unwanted AC line noise;
- the *60 Hz reference* is the noise reference obtained from the power line;
- the "*antinoise*" is the signal that is generated on the microcontroller through the LMS algorithm and created using pulse width modulation;

which may very well have high research, development, and marketing costs. Instead, we are referring to the "do-it-yourself" creation and implementation of active noise cancellation in situations where the purchase of high-cost commercial units is not feasible. It is the authors' hope that this device can be built from the specifications in this paper (either by students in an educational laboratory or by researchers on a budget) to gain capabilities once available only to those paying for a high cost commercial system.

- the *output*, *error*, or *clean signal* is the noise-free signal that is the end-result of the system. Although the goal is to have a noise-free result, the clean signal is used in the feedback loop and thus is confusingly called the error signal.

The microcontroller receives a 60 Hz reference signal (from the wall power line) and outputs a 180°-out-of-phase sinusoid. This "antinoise" is then added in analog to the contaminated signal, effectively subtracting the interfering 60 Hz component and eliminating the line noise in the contaminated signal. The feedback loop is then formed by inputting the cleaned signal back into the microcontroller as the error signal. The microcontroller uses this input to update the weights that change the amplitude and phase of the antinoise signal as needed to eliminate any remaining correlation. A block diagram of the overall system at a high-level is shown in Fig. 1.

2.2. Hardware Overview

The basic hardware components include an operational amplifier adder circuit, two RC-lowpass filters, a circuit to obtain the reference signal (voltage divider and bias circuit), and supporting circuitry for the microcontroller (Horowitz and Hill, 1989). The circuit schematic is shown in Fig. 3 (all resistors are 5% tolerance) and fits on a wallet-sized custom printed circuit board (PCB) shown in Fig. 2.

The device is powered with a 120 VAC transformer that outputs 12 VAC from which we derive the 60 Hz reference signal and 12 and 5 VDC power supplies.

The device is designed with the intent to minimize any spurious system noise during adaptive noise cancellation. On the board, a ground plane lies under all non-microcontroller circuitry to reduce spike noise that the microcontroller or other sources of broadband electromagnetic interference (EMI) might introduce into the system. Signal traces on the PCB are of minimum length to limit the total path length through the system and are kept far away from any power sources on the PCB to reduce the chance of reintroducing line-noise.

We use an Atmel ATmega32 microcontroller, which is capable of 8-bit operation at 16 MHz. Microcontroller capabilities needed for our implementation include:

- a pulse width modulation (PWM) channel for producing the antinoise signal

- two 8-bit (or better) analog-to-digital converter (ADC) inputs, to measure the reference and error signals
- two 8-bit timer/counters with separate prescalers and compare modes, to schedule input to and output from the microcontroller

The ATmega32's internal reference voltage is set to 2.56 V which means that the voltage range for inputs is 0–2.56 V. This requires the contaminated signal (and the reference signal) are biased to 1.25 VDC and are limited to ±1 V before input to the system.

2.2.1. Microcontroller Circuitry

The initial PC board design came from the eighth revision of a prototype board used for the ATmega32 in a classroom context.[4]

Power to the board is derived from a 12 VAC transformer connected to the AC line. We also use this power source to obtain a reference signal and thus require the power input to be AC. However, for the power circuit, we pass the AC signal through a general-purpose 1N4001 diode (rectifier) to obtain DC voltage. The DC voltage is then passed into a 5 V regulator (LM340LA2) which distributes power to the ATmega32. Included in this path are several small capacitors. The board also includes a 6-pin header to program this chip on the PC board and the ability for serial communications through an RS232 serial port for debugging purposes.

2.2.2. Adaptive Noise Cancellation Circuitry

The adaptive noise cancellation circuitry consists of circuits to obtain the 60 Hz noise reference, the analog adder, and two lowpass circuits.

The 12 VAC transformer output powers the microcontroller and also serves as the 60 Hz reference signal. The 12 VAC signal is passed through two stages: a voltage divider stage to bring the voltage range down to approximately ±1 V peak-to-peak and then a biasing circuit to bias the reference with a DC offset of approximately 1.25 VDC to meet the input requirements of the ATmega32 ADC. The values of the pulldown resistors are 10 kΩ, the pullup resistor in the voltage divider is a 100 kΩ, and the pullup resistor in the biasing circuit is a 81 kΩ.

[4]The latest version of the prototype board, schematics, and assembly instructions can be found at `http://www.nbb.cornell.edu/neurobio/land/PROJECTS/Protoboard476`

There is a 2.5 μF capacitor between the voltage divider and bias stages that creates a high-pass circuit to eliminate DC bias in the reference signal. The capacitance value is chosen such that the 3 dB cutoff frequency of the biasing circuit with the resistors seen in small signal parallel occurs at 7 Hz.

The analog adder is a standard unity gain inverting adder circuit followed by a unity gain inverting stage (overall non-inverting). The operational amplifiers are LM358 dual operational amplifiers in 8-pin DIP packaging. The input and feedback resistors are 20 kΩ. The inputs to the adder are the contaminated signal and the antinoise from the microcontroller PWM channel after it has been lowpass filtered with a simple RC filter. The output of the analog adder is the clean signal (system output).

Standard RC lowpass filters are used in two locations on the board. One lowpass filter is placed between the ATmega32's ADC and the error signal to eliminate possible high frequency content in the clean signal from aliasing down to the 60 Hz frequency and to comply with the sampling capabilities of the ATmega32's ADC. The second lowpass filter is placed between the output of the ATmega32's PWM output and the analog adder to smooth the signal into a sinusoidal waveform. Each RC lowpass circuit has a 20 kΩ resistor and a 0.04 μF capacitor to provide a 3 dB rolloff at 200 Hz.

A unique feature of our design and implementation is that the contaminated signal is never digitized—the entire signal path remains in analog. This allows for a much lower digitization rate and thus avoids loss of fidelity, delay, and sampling or reconstruction errors. The clean signal is sampled after low pass filtering in the feedback loop but only to modify the tap weights that control the antinoise. This does not effect the output signal bandwidth.

2.3. Software Overview

The software is responsible for generating an antinoise signal and making realtime adjustments based on both the reference 60 Hz noise and the error signal.[5]

An optimized LMS algorithm minimizes correlation between the reference and error signals by adjusting weights $W_1, W_2, \ldots, W_N$. These comprise a finite impulse response (FIR) filter on past reference inputs $R_1, R_2, \ldots, R_N$ to produce an antinoise output, `antinoise` which is summed in analog to

[5] For those wishing to build the device on their own, our microcontroller code is available at `http://www.nbb.cornell.edu/neurobio/land/Papers/AdaptiveNoiseControl`.

the contaminated input to produce the system output, which we feed back in as our error measurement, `error`.

Our software is structured into three primary components:

1. a timer interrupt that writes `antinoise` to PWM output at full clock speed with an interrupt triggered on counter overflow;
2. a timer interrupt with a clk/64 prescaler that performs ADC, alternatively storing the a measurement of the reference source R_n in a circular buffer, and storing the instantaneous system error, `error`;
3. finally, a main loop polls for state information which is set during the second timer interrupt and either updates filter weights when there is a new error value, or computes the next value of `antinoise` when a new value from the reference input is read.

2.3.1. Generating the antinoise

We implemented a microcontroller-based, fixed-point adaptive FIR filter, so we compute output by summing the product of weights and previous values of the reference input:

$$\texttt{antinoise} = \sum_{i=1}^{N} W[i]R[i] \tag{1}$$

To avoid an undesired high-frequency harmonic, we output our signal pulse width modulated at the maximum rate supported by the ATmega32: 62,500 Hz. We do this through an interrupt, which linearly interpolates between desired outputs. Since the sample rate (see below) and the output rate are related by a power of two, we can find an appropriate increment for the interpolation with a simple right shift.

2.3.2. Filter Characteristics

Since we are targeting 60 Hz noise (although this can be modified for 50 Hz line noise), we can optimize our filter for that purpose. We have a filter length of eight, at a sample rate for the input of 488 Hz, which gives an ideal window size of $488/8 = 61$ Hz. This means that each sample is 45° apart in the 60 Hz reference signal, producing ideal conditions for adapting.

In order to eliminate any possible aliasing in the error signal, we need to sample at a rate consistent with the Nyquist theorem. Since our cancellation focused on 50–70 Hz signals, our input sampling frequency needs to be at least 140 Hz, and we need to lowpass the contaminated signal.

2.3.3. Adapting Weights

We are using an optimized LMS algorithm based on Stefano et al. (2005) where weights are updated such that:

$$W_{n+1} = W_n + \mu \cdot \texttt{error} \cdot R_n \tag{2}$$

A further optimization that we employ replaces `error` with its sign (Stefano et al., 2005):

$$W_{n+1} = W_n + \mu \cdot \text{sign}(\texttt{error}) \cdot R_n \tag{3}$$

By achieving a small μ through adding the full R_n value to the fractional section of the weight, W_{n+1}, (see Sec 2.3.4), we can eliminate all of the computationally-costly multiplications from the filter convergence, so that it runs on a low-speed microcontroller.

2.3.4. Fixed-point Math

Floating point arithmetic is not practical with the computational limitations of a low-cost microcontroller and the speed required in a realtime system. We used a 16:16 fixed-point system, which devotes 16 bits to a signed integer component of the number, and 16 for the fractional component. Since all of our inputs were coming in from the internal ADC in eight bits, we only need define fixed-point multiplication with an eight-bit unsigned integer (ie. 8:0 fixed-point).

3. Results

We present four tests to demonstrate the abilities and effectiveness of our adaptive noise cancellation (ANC) system. The first three tests are aimed at demonstrating the system's abilities through somewhat contrived examples whereas the fourth is a simulation of a neurobiological laboratory experiment.

3.1. Pure Frequency Signal Test

To construct the pure frequency signal test, a single frequency signal is added to a 60 Hz noise signal to form the contaminated signal. The voltages of the two components can be varied to achieve the desired signal to noise ratio (SNR) for testing and evaluation. The goal is to have the pure single frequency as system output without the 60 Hz signal.

As an example, a 0.8 V peak-to-peak 200 Hz signal is added to a 0.8 V peak-to-peak 60 Hz signal to create the contaminated signal with a SNR equal to 1. The contaminated signal and 60 Hz reference are system inputs. The ANC system input and output over time is shown in Fig. 4. The top of the figure shows the contaminated signal and the bottom shows the clean signal (output) of the system after convergence. The frequency domain view of the cancellation is shown in Fig. 5 and allows us to quantify the cancellation. The top shows the FFT of the contaminated signal, and the bottom shows the FFT of the clean output. There is about 25 dB of cancellation of the 60 Hz component.

Note that the system is designed to adapt to various amplitudes of 60 Hz line noise equally well (ie. over all amplitudes of noise). However, the system does have suppression level limits that are functions of the microcontroller. First, the microcontroller has a input range of 0–2.56 VDC. Because input signals are biased to approximately 1.25 VDC, the line-noise contaminated signal cannot exceed the approximately 1.25 V available range. The upper bound on the voltage range of cancellation is therefore limited to approximately 1.25 V peak amplitude. Second, the microcontroller has an 8-bit analog digitizer which limits resolution to about 1/256 of full scale or about -24 dB. Of the two factors, the latter is likely the limiting factor in our results since the contaminated test signal voltage range falls within the acceptable input specifications.

3.2. Convergence Test

The convergence rate of the system—how fast it can achieve suitable cancellation—is also measured. The convergence rate is controlled by the software variable μ, which controls step size. With a moderate setting of μ, the system was stable and converged within three seconds. Convergence for AC line noise alone is shown in Fig. 6.

3.3. AC Line Spectrum

We also put pure AC line noise into our system to determine how well the system reduced noise alone. Fig. 7 shows the AC line spectrum before (top) and after (bottom) cancellation. Cancellation of the fundamental 60 Hz signal is reduced by more than 25 dB. A typical notch filter implementation achieves only about 15 dB (Ramos et al., 2007).

3.4. Action Potential Test

Although the first examples are not likely to be encountered in the laboratory, they are useful to obtain a sense of the overall performance metrics of our ANC system. A simulation of a neurobiology experiment is now presented.

A typical laboratory experiment might be to measure an action potential—an electrical signal generated and transmitted by nerve cells. Because of the power line noise in laboratories and the low voltage measurements trying to be performed, the 60 Hz noise can easily overwhelm the action potential. In order to simulate these conditions, we wrote a script to generate random action potentials of simulated electric organ discharge and then summed this with a 60 Hz signal from the AC line at a SNR of about 3/2 (top of Fig. 8). After the system converged, action potentials without noise are recovered (bottom of Fig. 8).

4. Discussion

The noise cancellation system described is a low-cost alternative to commercial units (Scientific, 2008). Using low-cost commodity microcontrollers and analog summation, the device produces 20–25 dB of cancellation of a pure 60 Hz sine wave while giving an analog bandpass of greater than 50 kHz. It can easily be modified to cancel 50 Hz with a software change to the sample rate.

The limitation of our system is that only the primary 60 Hz harmonic is canceled. Second and third harmonics are attenuated up to 3 dB, and spike noise and other line-correlated noise cannot be canceled. The limitation is due to the relatively low computational rate of the microcontroller which limits the length of the FIR filter. However this limitation is not important in most lab situations. The distortion standard for line power (IEEE-519) specifies that the total harmonic distortion should be about 26 dB below the fundamental including up to the 11th harmonic. Measurements in our electrophysiology laboratory (late 1970's design with 3-phase 480 volt motors for air handling) showed 3rd and 5th harmonics to be 24–30 dB below the 60 Hz fundamental at various times with very small 2nd and 4th harmonics. Measurements in a student electronics lab showed almost equal 2nd and 3rd harmonics at about -21 dB. Our circuit essentially removes the 60Hz fundamental by 20–25 dB and attenuates the 2nd and 3rd harmonics up to 3 dB.

We foresee use of this system in student laboratories and other situations where low cost and adequate performance are important. Students should certainly be taught how to debug and minimize noise situations (e.g. multiple grounds and use of Faraday shielding) but should also see that active techniques can result in faster setup times and cleaner results. The system is adequate to 'clean up' spike trains contaminated by 60 Hz with twice the amplitude of the spike train. Simulated spikes (but real AC noise) at a SNR equal to 1, which would render a spike threshold measurement useless, become clean enough to resolve 10% differences in spike amplitude.

Acknowledgments

We wish to thank Cornell University for the generous use of their teaching facilities and laboratory supplies. Additionally, we want to thank the developers of the free software AVR-GCC compiler and AVRDUDE programmer, whose tools were indispensable. We also thank Ben Arthur, Boris Chagnaud, Aaron Rice, and the anonymous reviewers for their thoughtful comments and revisions.

References

M. L. Ahlstrom and W. J. Tompkins. Digital filters for real-time ECG signal processing using microprocessors. *IEEE Transactions on Biomedical Engineering*, 32(9):708–713, September 1985.

Ahmen Elhossini, Shawki Areibi, and Robert Dony. An FPGA Implementation of the LMS Adaptive Filter for Audio Processing. In *Proc. IEEE International Conference on Reconfigurable Computing and FPGA's*, pages 1–8, September 2006.

M. Ferdjallah and R.E. Barr. Adaptive digital notch filter design on the unit circle for the removal of powerline noise from biomedical signals. *IEEE Transactions on Biomedical Engineering*, 41(6):529–536, June 1994.

P. Horowitz and W. Hill. *The Art of Electronics*. Cambridge University Press, New York, NY, USA, 1989.

Sen M. Kuo and Dennis R. Morgan. *Active Noise Control Systems: Algorithms and DSP Implementations*. Wiley Series in Telecommunications and Signal Processing. Wiley & Sons, 1996.

P. A. Lynn. Online digital filters for biological signals: some fast designs for a small computer. *Medical and Biological Engineering and Computing*, 15 (5):534–540, September 1977.

Rafael Ramos, Antoni Mànuel-Làzaro, Joaquín Del Rìo, and Gerard Olivar. FPGA-Based Implementation of an Adaptive Canceller for 50/60-Hz Interference in Electrocardiography. *IEEE Transactions on Instrumentation and Measurement*, 56(6):2633–2640, December 2007.

Quest Scientific. Hum Bug Noise Eliminator, 2008. URL `http://www.digitimer.com/research/quest/`. Promotional Material.

Antonio Di Stefano, Alessandro Scaglione, and Costantino Giaconia. Efficient FPGA Implementation of an Adaptive Noise Canceller. In *International Workshop on Computer Architectures for Machine Perception*, pages 87–89, July 2005.

Figures

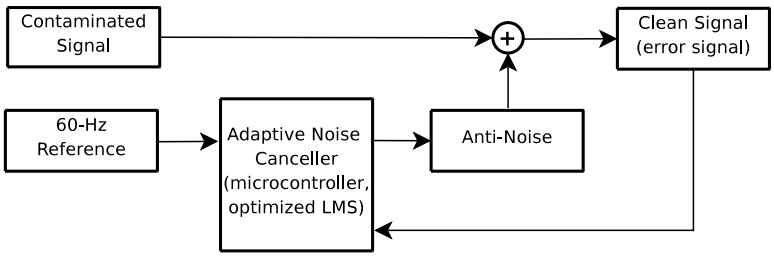

Figure 1: High-level block diagram overview.

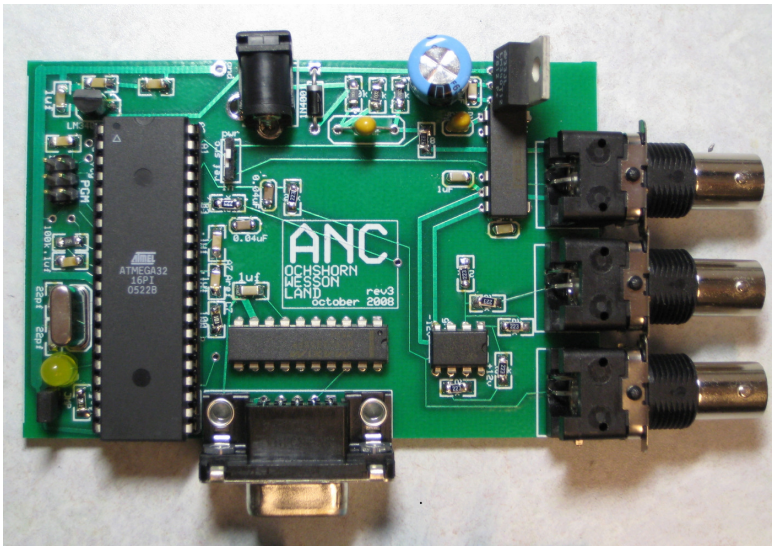

Figure 2: The PCB implementation of the adaptive noise cancellation system.

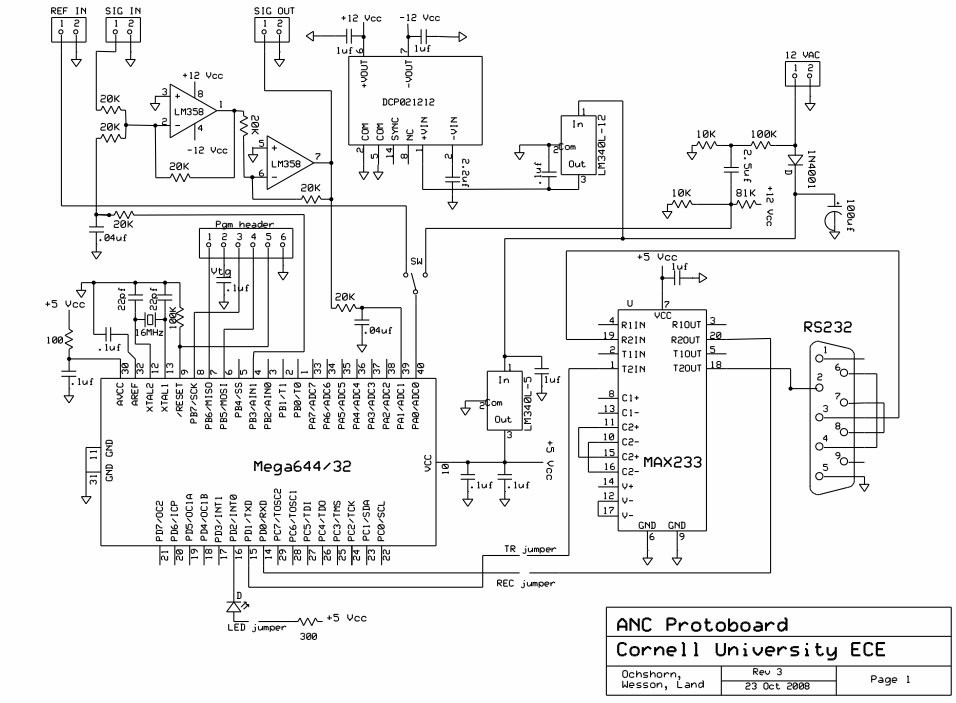

Figure 3: The PCB schematic.

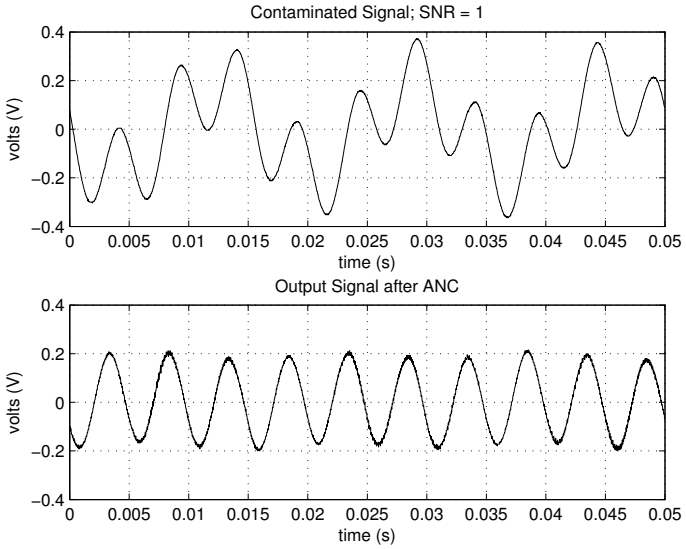

Figure 4: Time domain view of pure tone cancellation.

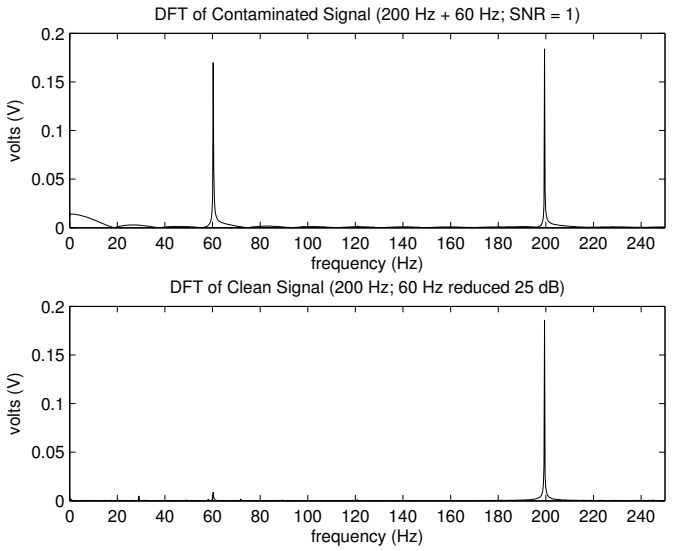

Figure 5: Frequency domain view of pure tone cancellation.

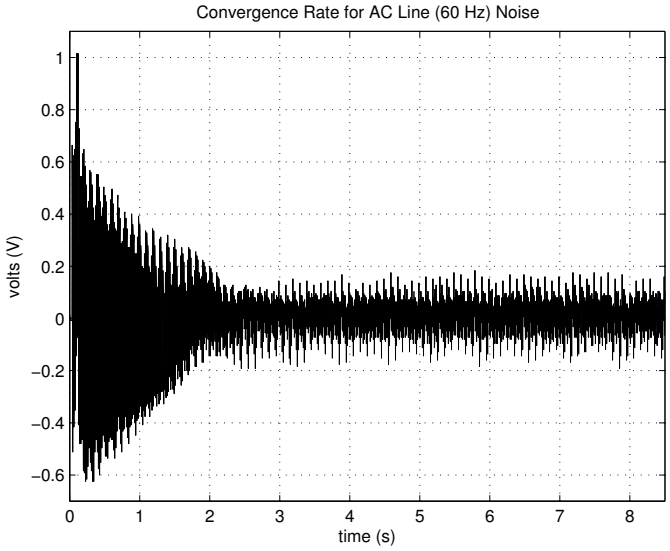

Figure 6: Convergence rate for pure AC 60 Hz line noise. Convergence occurs in just over two seconds.

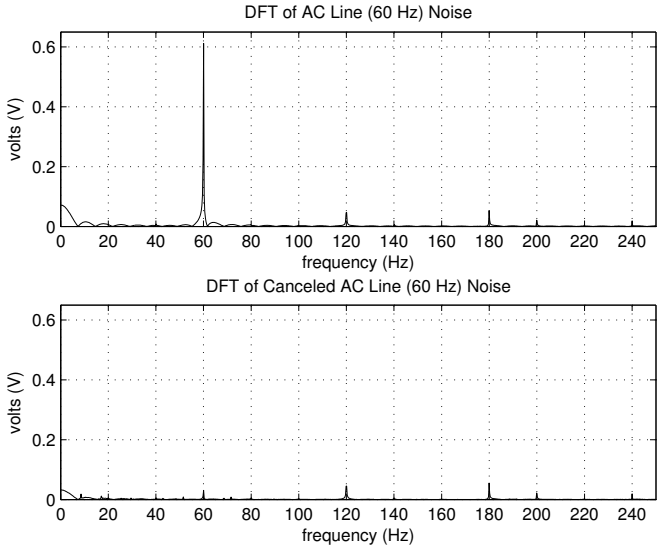

Figure 7: Frequency domain view of pure AC line 60 Hz noise cancellation. The fundamental 60 Hz component is reduced 25 dB.

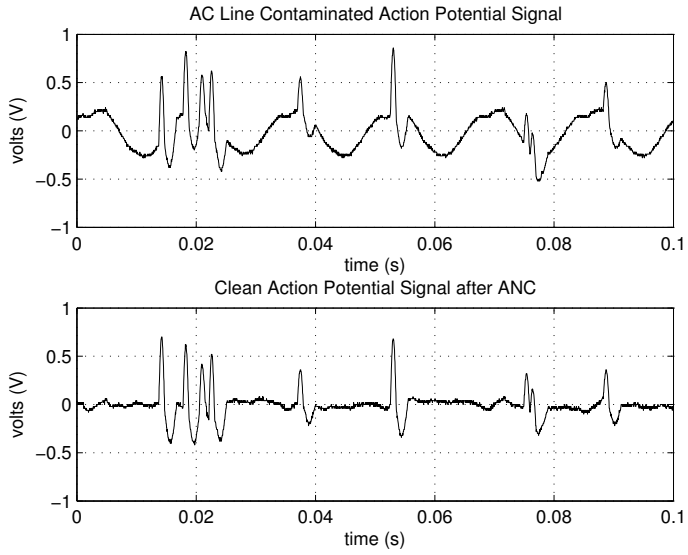

Figure 8: Action potential test demonstrating the results of a simulated neurobiology laboratory experiment. Top shows contaminated signal and bottom the clean signal.