How Noise-Cancelling Headphones Create Silence in Microseconds

- need outer and inner microphone
- focus on predictable, continuous, constant noise
- can't really cancel anything unpredictable

Active Noise Cancelling - From Modeling to Real-Time Prototyping

- focus on simulation first, verify algorithm behavior then move to real-world testing

Audio amp classes as fast as possible

- will prolly want a class D amp. + 1-bit DAC

FxLMS PDF Notes

Filtered-x Least Mean square fourth algorithm active noise control

- main drawback of FxLMS is use of fixed-step size (creates compromise between noise reduction performance and convergence speed of the ANC sys.)

[redacted]

- $e(n) = d(n) - y(n)$
- $y(n) = W^T(n)x(n)$

[redacted]

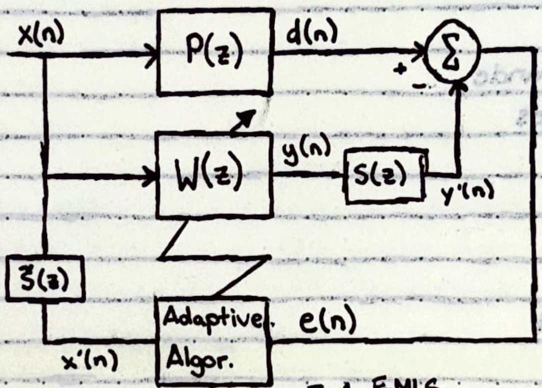


Fig 1. FxMLS

Legend
- $d(n)$ - primary noise to be cancelled
- $P(z)$ - primary path TF
- $S(z)$ - secondary path TF
- $W(z)$ - adaptive filter
- $y(n)$ - adaptive filter response
- $e(n)$ - residual error

Misc. Notes on Adaptive Filters/FxLMS Algo.

- gen. idea is that one will adjust filter weights in an effort to minimize error
- i.e. when step size too small it will take too long to converge/ too big you won't get good noise cancellation
- since ANC works mainly w/ white or gaussian noise we can converge quite quickly, making ANC possible
- convergence is μ and must have an upper bound $(0 < \mu < 2/\lambda_{max})$ where λ_{max} is the largest eigenvalue of autocorrelation matrix

$R = E(x(n)x^H(n))$ max. conv. speed $\rightarrow \mu = 2/\lambda_{max} + \lambda_{min}$

- for white noise: $R = \sigma^2 I$ where σ^2 is variance of the signal

Diff. Adaptive Filtering Notes

- we create the performance function by assuming d_k, e_k, x_k are all zero mean, wide-sense stationary and statistically independant
- tap length is important for algorithm
- $W_{k+1} = W_k - \mu R^{-1} \nabla$

- first determine equation to characterize the error: $e(n) = d(n) - y(n)$
- $\xi^2 = \mathbb{E}(e_k^2) \rightarrow R = \mathbb{E}(X_k X_k^T)$ & $P = \mathbb{E}(d_k X_k)$ $\qquad e_k^2 = d_k^2 + W^T X_k X_k^T W_k - 2 d_k W_k^T X_k$
- R is an L×L matrix & P is an L×1 vector, therefore

$$\xi^2 = \mathbb{E}(e_k^2) = \mathbb{E}(d_k^2) + W^T R W - 2P^T W$$ (notation explained below)

Note: adaptive filter coeffs. are manipulated to minimize the statistically-averaged squared value of the error signal ξ^2. Given then, the funct. is a quadratic funct. of the weight vector W. This means that there is only one single min. value of ξ^2 over all coeffs. in W

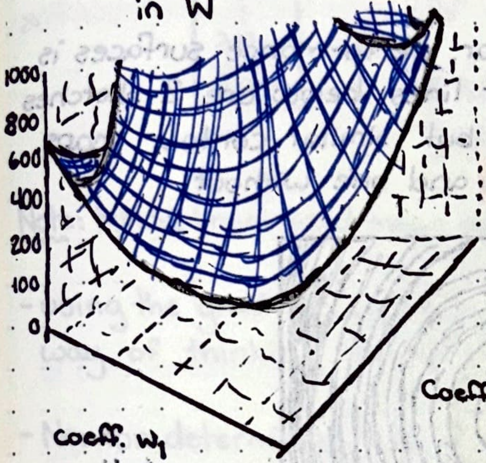


Fig 2. Crude Performance Surface

- In the picture to the left, we have an adaptive filter w/ two weight variable coefficients (w_0 & w_1), more specifically its "performance surface" or a plot of its MSE (ξ^2)(Mean ~~Standard~~ square Error) as a funct. of w_0 & w_1

- It creates a bowl-shaped "hyperparabola", the minimum of this surface is the best possible value for the filter under consideration, there are different ways of extracting the min.

- In principle, the shape of the "performance surface" depends on the specific optimal values of the coeffs. for a specific/particular applicat., the correlations between the components of the observation vect. and desired signal (X_k and d_k respectively) and especially the correlations of the components of the observations X_k w/ each other,

* Note: the eqs. are taken from an adaptive linear combiner (ALC)(below) whose output can be desc. by equation $y_k = \sum_{l=0}^{L} x_{lk} w_{lk}$ where w_{lk} represents the variable weights of the filter and x_{lk} is the signals that are inputs to those weights

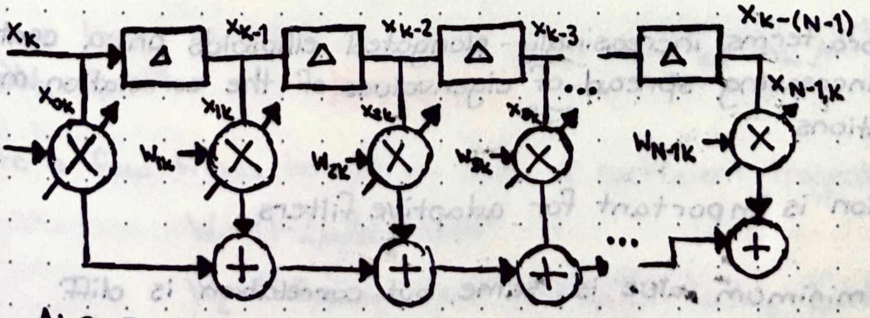


Fig 3. ALC Block Diagram

Secondary Note: adaptive sys. use diff. notational conventions, first subscript denotes the ind. filter coeff. under cons. and second ind. time ind.

Adaptive Filter Notes Cont...

- the observation vector (set of signal inputs to var. weights),
- if the adaptive processor is the form of FIR, the obs. vector is made of components (x_{lk}) that are equal to delayed versions of the original inputs where $x_{lk} = x_{k-1}$, which approx. n'_k
- frequently the inputs are expressed as column vectors as well as the variable filter coeffs.

$$X_k = \begin{bmatrix} x_{0k} \\ x_{1k} \\ x_{2k} \\ x_{3k} \\ \vdots \\ x_{L-1,k} \end{bmatrix} \quad \& \quad W_k = \begin{bmatrix} w_{0k} \\ w_{1k} \\ w_{2k} \\ w_{3k} \\ \vdots \\ w_{L-1,k} \end{bmatrix}$$

- this gets us to finally, $y_k = \sum_{l=0}^{L-1} x_{lk} w_{lk} = X_k^T W_k = W_k^T X_k$ where T means matrix transpose operator
- another way of char. performance surfaces is in terms of contour lines, below are 2 sketches showing 2 different but similar contour maps one w/ correlation and one without

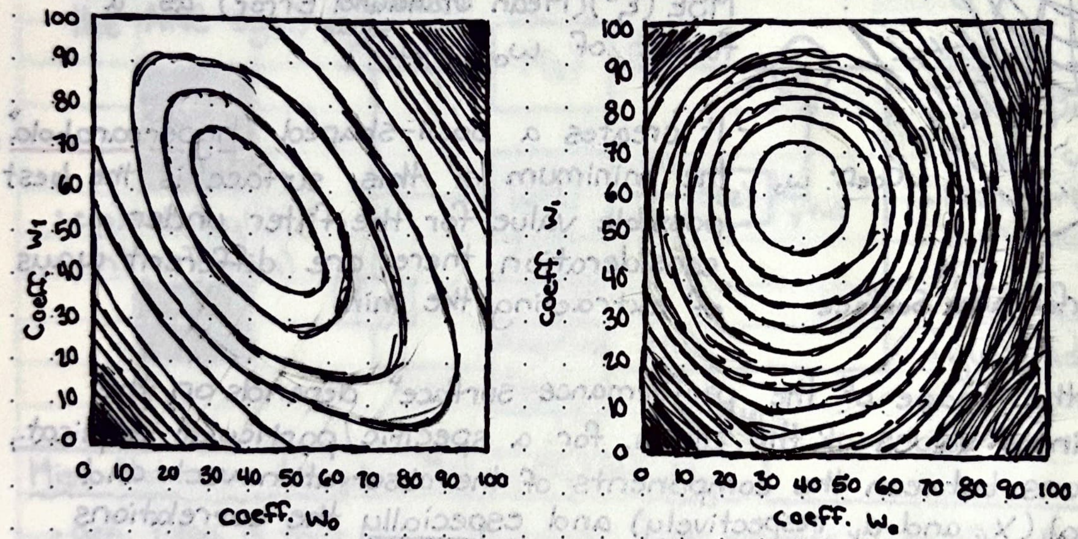


Fig. 4: Very Rough Contour plots

- In the right figure above the correlation between x_0 & x_1 is zero ($\rho = \emptyset$) and on the left correlation between x_1 & x_2 is 0.7 ($\rho = 0.7$). This is because the performance surface has circular contour lines if the values are uncorrelated. They become increasingly, ellipsoid and elongated as the correlation increases.

Note: In linear algebra terms, increasingly-elongated ellipsoids on a contour plot reflect increasing spread of eigenvalues of the correlation matrix of the observations

↑ review concept

- Degree of correlation is important for adaptive filters
- In image above minimum value is same, but correlation is diff

For Finding MSE Analytically:

- If only one weight (ω) was used, minimum ξ^2 would be found by differentiating w/ respect to ω, setting derivative equal to zero, and solving for w

- If instead vector W is used, employ a similar set of operations but using a gradient operator

$$\nabla = \frac{\partial \xi^2}{\partial W} = \begin{bmatrix} \frac{\partial \xi^2}{\partial w_0} \\ \frac{\partial \xi^2}{\partial w_1} \\ \vdots \\ \frac{\partial \xi^2}{\partial w_{L-1}} \end{bmatrix}$$

- we also know that $\xi^2 = \mathbb{E}(d_k^2) + W^T R W - 2P^T W$

- after differentiating w/ respect to elements of W we get $\nabla = 2RW - 2P$

- if W^* minimizes ξ^2 then it is found by $\nabla = \emptyset = 2RW^* - 2P \rightarrow W^* = R^{-1}P$

Note: MMSE (minimized mean ~~standard~~ square error) depends upon only R & P

- using the gradient equation and multiplying by $\frac{1}{2}R^{-1}$ we can get a new way of thinking of W^*: $W^* = W - \frac{1}{2}R^{-1}\nabla$

- Newton determined that if we have perfect knowledge of R we can move "to the bottom of the bowl" in one step because we know both position and slope (surface is quadratic & stats are known)

- this doesn't happen in real life so we must estimate R & ∇ empirically, one way to do this is to move down in small increments, this done w/ the relationship: $W_{k+1} = W_k - \mu R^{-1}\nabla$

- this approach emphasizes finding the direction "down" the performance surface and moving in steps (μ) towards the bottom

*Note: Larger step size (μ) is faster for moving to the bottom faster, but as the bottom approaches residual error would be higher since the large step size can cause coefficients to oscillate around their true values

- Now, we know $\nabla = 2RW - 2P$ which gives us: $W_{k+1} = W_k - \mu R^{-1}(2RW - 2P) = (1-2\mu)W_k + 2\mu W^*$

- Iterate a few times to get an idea of coefficient trajectories: (W_0 start)

$$W_1 = (1-2\mu)W_0 + 2\mu W^*$$

$$W_2 = (1-2\mu)W_1 + 2\mu W^* = (1-2\mu)[(1-2\mu)W_0 + 2\mu W^*] + 2\mu W^*$$

$$= (1-2\mu)^2 W_0 + 2\mu W^*(1 + (1-2\mu))$$

$$W_k = (1-2\mu)^k + 2\mu W^* \sum_{i=0}^{k-1} (1-2\mu)^i$$

Adaptive Filters Cont...

- using formula of finite sum of exponentials: $W_k = (1-2\mu)^k W_0 + W^* \frac{1-(1-2\mu)^k}{1-(1-2\mu)}$
$$= (1-2\mu)^k W_0 + W^*(1-(1-2\mu)^k)$$
- can also be written in form: $W_k = e^{-t_k/\tau} W_0 + (1-e^{-t_k/\tau})W^*$
  where $\tau = 1/2\mu$

The Least Mean Squares (LMS) Adaptation Algorithm

- moving from the determined gradient descent equation: $W_{k+1} = W_k - \mu R^{-1} \nabla$
- this method has two main problems:
  - we don't know the value of statistics for R & P
  - don't know value of ∇
- to avoid these issues we do the following:
  - ignore R^{-1} in the above equation
  - replace average stats. of R & P w/ their instantaneous values
- this creates the equation of: $W_{k+1} = W_k - 2\mu(X_k X_k^T W_k - d_k X_k)$
- using previous defs. of y_k & e_k we get: $W_{k+1} = W_k + 2\mu X_k e_k$
- this replaces the ideal trajectories from before w/ noisy approximations, this is why it is sometimes called a "stochastic gradient" algorithm
- the algorithm while quite effective does suffer from potentially slow convergence when the components of the observation vector are highly correlated since the algorithm produces tracks perpendicular to the contour lines and this is not the fastest way down the slope w/ highly correlated signals
- there have been several proposed ways to deal w/ correlated signal observation
  1. Use other adaption algorithms (RLS or recursive least mean squared)
  2. Alternate filter structures (FIR lattice struct)
  3. Use frequency domain approaches
  4. Use subband approaches

Reason for FxLMS instead of LMS

- if one attempts to implement LMS algorithm on practical ANC, the secondary path of the antinoise is complicated by traveling to the error microphone (digital signal → analog signal → digital again)
- this introduces frequency and phase distortions known cumulatively as $S(z)$ which includes D/A convertor, reconstruction filter, amplifier, loudspeaker, acoustic path from loudspeaker to mic (error), pre-amp, anti-aliasing filter, & A/D convertor
- to counteract ($S(z)$) we place an inverse filter between ref. mic and LMS algorithm which allows algorithmic convergence
- easiest way to determine ($S(z)$) is to use LMS w/ white noise as an input

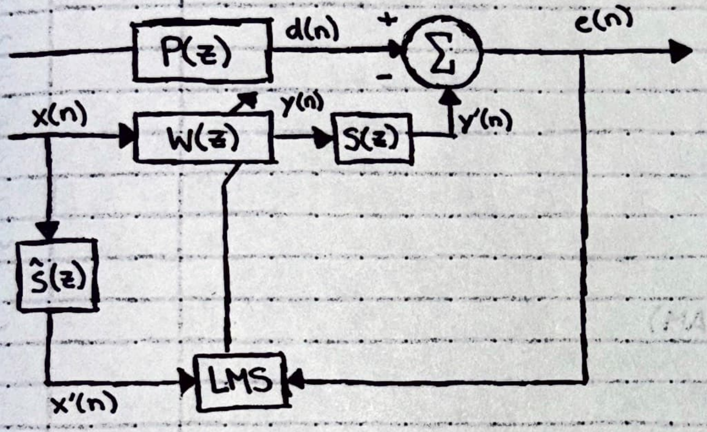
Why FxLMS? Cont...

- the inverse impulse response is found by bringing impulse response of $S(z)$ into freq. domain, taking its inverse and bringing it back to time domain

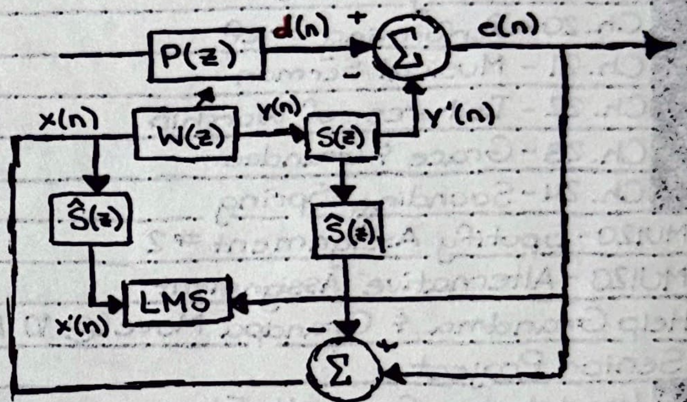
$x'(n)$ = convolution of $\hat{S}(n)$ and $x(n)$

An Advanced Feedback Filtered-x Least Mean Square Algorithm...

- most everyday sounds that constitute noise pollution are low frequency (<1kHz)
- in feedforward ANC a correlated noise input is sensed by a second microphone before propogating past a secondary source
- in feedback ANC, the secondary signal is predicted and is not generated using a secondary microphone



Feedforward ANC Network



Feedback ANC System

Active Noise Cancelling Using Filtered-xLMS & Feedback ANC Filter Algorithms

- hybrid microphone on inside and outside of headphones?
- does the FxLMS algorithm always run the filter or does the refresh rate of the algorithm update vary from product to product
- can you have an input microphone array

- LMS is constantly updating but first convergence is longer since weights must be changed by more initially
- the constant LMS is most assuredly a power consideration
- find one good omnidirectional mic instead of input microphone array
- find a microphone w/ good frequency response within range of specified frequency band
- 3D printer setup should be done this weekend
- verify chosen headphone driver has the correct frequency response
- class D or class AB amplifier

Seth & Andrew Senior Project Weekly Breakdown

Week One (Redux) (2/23-2/28)	Assignee	Status
FxLMS Alg. Simulation	Andrew	
LTSpice Preamp & Anti-Aliasing Filter Simulation	Seth	
Altium PCB Rough Draft	Seth	
Split .stl Files Efficiently	Seth	
Circuitry Catalog	Both	
Begin Detailed Design Review Document	Both	

Low-Pass Filter & Anti-Aliasing Filter Notes

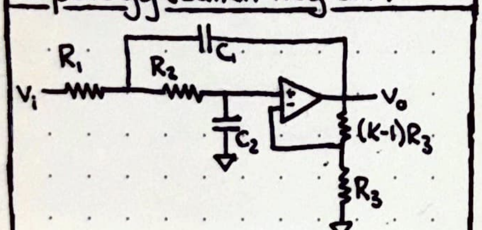
- will need a low-pass filter w/ a cutoff frequency of around 1 kHz
- must be a relatively low-order analog filter
- butterworth filter?
- ~~b~~ perhaps Chebyshev LPF
- oversampling and pre-amp

[redacted]

LTSpice Notes:

- LM358 for op-amp?
- similar to?
- used LT1001
- TI uses LPF w/ sallen-key architecture

Topology (Sallen-Key LPF)



- cutoff freq. = 1 kHz
- secondary ckt for gain
- need output range of mic
- input range of codec

$Q=\frac{\sqrt{C_1C_2}}{C_1+C_2}\sqrt{\frac{R_2}{R_1}}$ $\omega_0 = 1/\sqrt{C_1C_2R_1R_2}$

$2\pi f_c = 1/\sqrt{C_1C_2R_1R_2}$ $C_1=C_2=2\text{nF}$

$1=\frac{\sqrt{4\times10^{-18}}}{4\times10^{-9}}\sqrt{\frac{R_2}{R_1}}$ (HIGH PASS EQs ooops)

$\downarrow$

$1=0.5\sqrt{\frac{R_2}{R_1}}$

$2000\pi = 1/\sqrt{4\times10^{-18}R_1R_2}$

$4=\frac{R_2}{R_1}$

$2000\pi = 1/\sqrt{4\times10^{-18}5R_1}$

$4R_1=R_2$

$R_1=1.26651\times10^9\,\Omega$ (Too LARGE!)

$C_1=C_2=20\text{nF}$ ✗

[redacted]

$C_1=C_2=0.2\,\mu\text{F}$

$R_1=126.651\text{ k}\Omega \rightarrow 127\text{ k}\Omega$

$R_2=506606\text{ k}\Omega \rightarrow 511\text{ k}\Omega$ or $499\text{ k}\Omega$

$2=\frac{\sqrt{R_1R_2}}{R_1+R_2}$ $\frac{2}{1591.55}=1/R_1+R_2$

$795.775=R_1+R_2$

$2000\pi = 1/\sqrt{4\times10^{-14}R_1R_2}$

$=R_1+\frac{633257}{R_1}$

$633257=R_1R_2$

$\frac{633257}{R_1}=R_2$

[redacted]

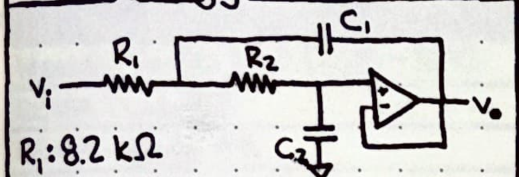
[redacted] $Q = \sqrt{R_1 R_2} / R_1 + R_2$ $f_c = 1\text{ kHz}$

$Q = 1$ $2 = \sqrt{\frac{C_1}{C_2}} \rightarrow 4 = \frac{C_1}{C_2} \rightarrow 4C_2 = C_1$

Maybe? $R_1 = R_2 = 10\text{ k}\Omega$ $2000\pi = 1/\sqrt{C_1 C_2\, 10k \cdot 10k}$ ✗ Caps too small

$C_1 = C_2 = 0.2\ \mu F$

Final Topology Used



R_1: 8.2 kΩ
R_2: 8.2 kΩ
C_1: 20 nF
C_2: 20 nF

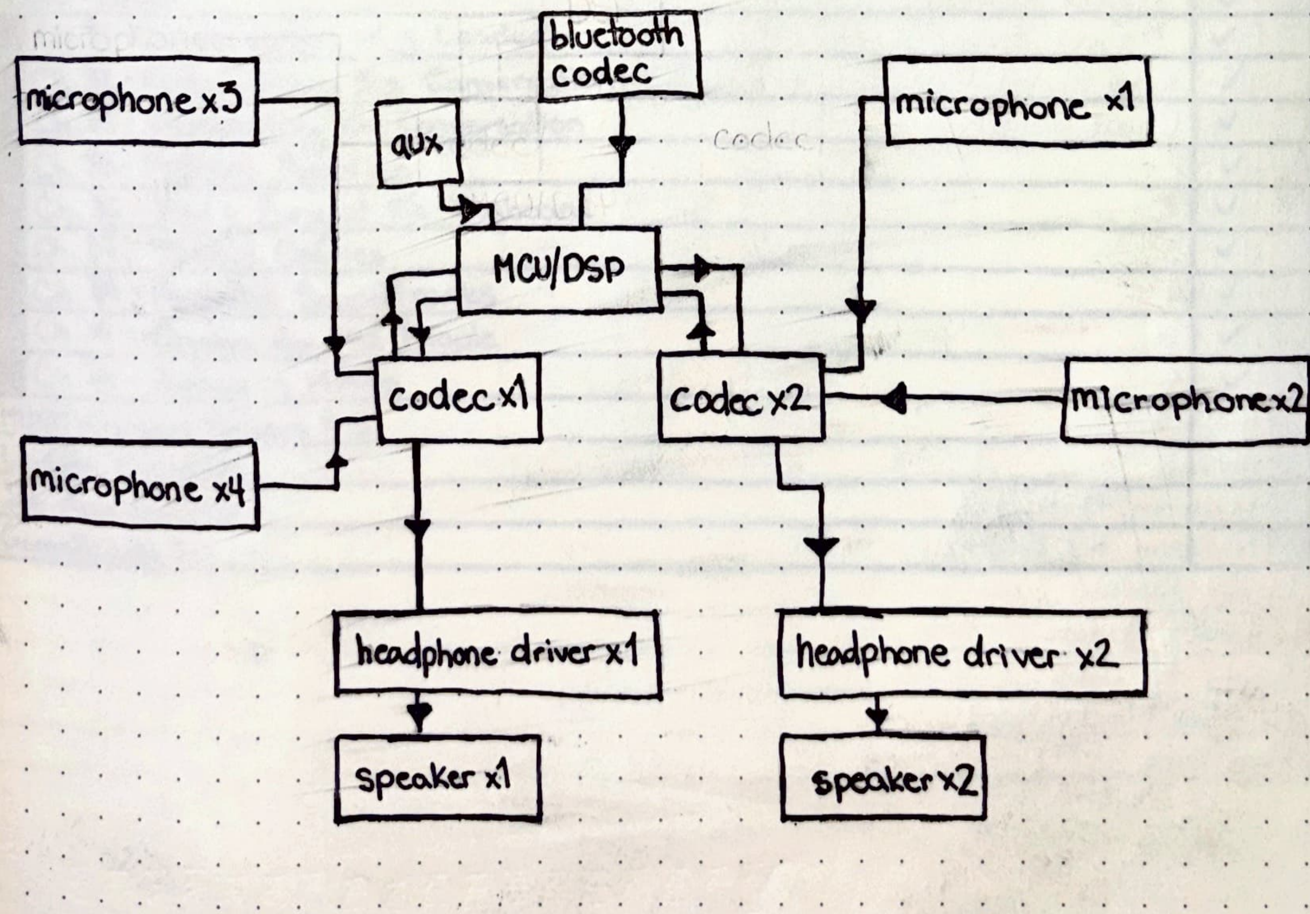
PreAmplifier Notes

microphone freq. range → 100 Hz – 20,000 Hz
audio codec input →

Questions For Dr. Pilkington

- ~~audio codec provides all we need & do we still want to make our own anti-aliasing filter (is it necessary)~~

Rough Block Diagram



Daily Checklist
• Senior Proj.
• EE459?
• DSP Cheat Sheet Redo

Schematic 2 Tabulation

Part	Label	Value	Amount
Res.	R5	1kΩ	5
Res.	R6	1kΩ	6
Res.	R7	1kΩ	7
Res.	R8	1kΩ	8
Cap.	C22	0.1µF	10
Res.	Rp3	10kΩ	3
Res.	Rp4	10kΩ	4
Cap.	C23	0.1µF	11
Cap.	C24	1µF	8
Cap.	C25	0.1µF	12
Cap.	C26	1µF	9
Cap.	C27	0.1µF	13
Cap.	C28	0.1µF	14
Cap.	C29	1µF	10
Cap.	C30	0.1µF	15
Cap.	C31	1µF	11
Cap.	C32	10µF	2
Cap.	C33	0.1µF	16
Cap.	C34	1µF	12
Cap.	C35	0.1µF	17
Cap.	C36	0.1µF	18
Cap.	C37	0.1µF	19
Cap.	C38	0.1µF	20
BT	IC2	n/a	n/a

EE461 Schematic Tabulation

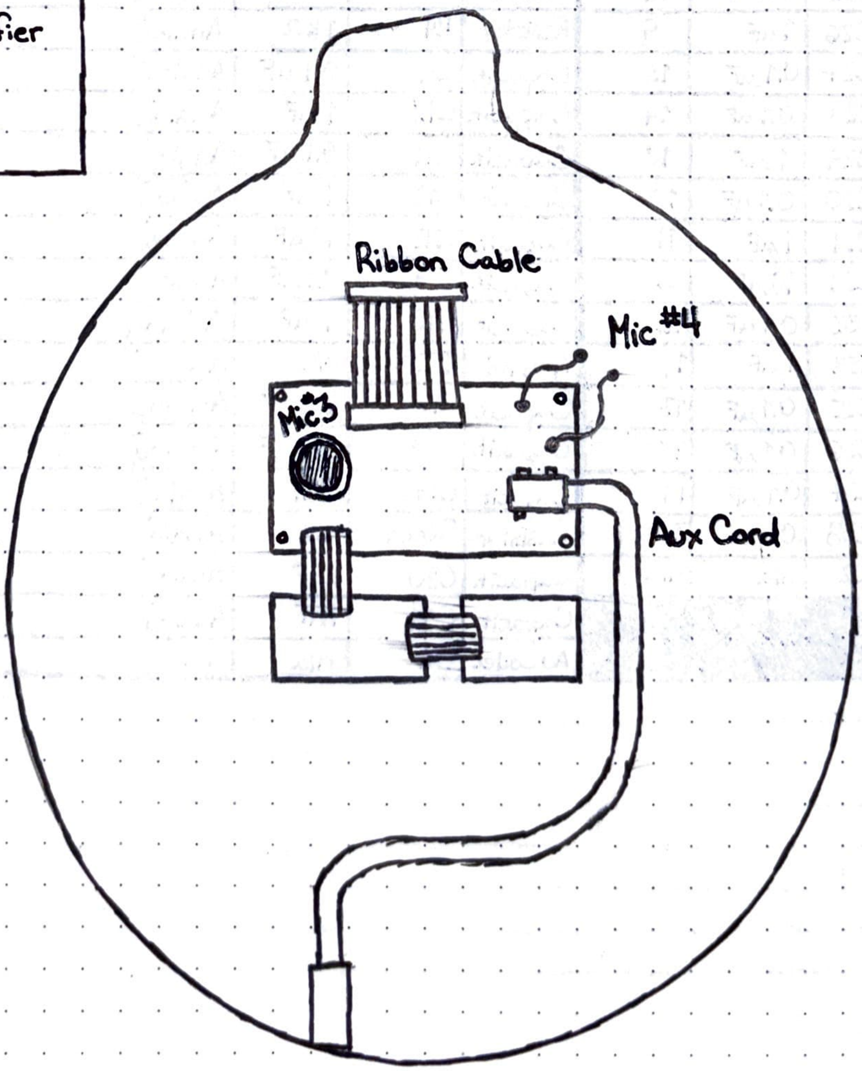
Part	Label	Value	Analog/Digital Connection	Number
AuCodec1	U1	n/a	n/a	1
R_{p1} ↔	Resistor	10kΩ	[redacted] Digital	1
R_{p2} ↔	Resistor	10kΩ	[redacted] Digital	2
Capacit.	C1X	220µF	Analog	1
Capacit.	C2X	220µF	Analog	2
Aux. Jack	J1	n/a	Analog	1
Capacit.	C1	0.1µF	Digital	1
Capacit.	C2	1µF	Digital	1
Capacit.	C3	0.1µF	Digital	2
Capacit.	C4	1µF	Digital	2
Capacit.	C5	0.1µF	Analog	3
Capacit.	C6	0.1µF	Analog	4
Resistor	R1	1kΩ	Analog	1
Resistor	R2	1kΩ	Analog	2
Capacit.	C7	0.1µF	Analog	5
Capacit.	C8	0.1µF	Analog	6
Resistor	R3	1kΩ	Analog	3
Resistor	R4	1kΩ	Analog	4
Capacit.	C9	0.1µF	Analog	7
Capacit.	C1Ø	1µF	Analog	3
Capacit.	C11	0.1µF	Analog	8
Capacit.	C12	1µF	Analog	4
Capacit.	C13	10µF	Analog	1
Capacit.	C14	0.1µF	Analog	9
Capacit.	C15	1µF	Analog	5
Amplif.	IC1	n/a	n/a	n/a
Capacit.	C17	2.2µF	Analog	1
Capacit.	C18	2.2µF	Analog	2
Capacit.	C19	1µF	Analog	6
Resistor	RGain	???	Analog	
Capacit.	C20	1µF	Analog	6
Capacit.	C21	1µF	Analog	7
AuCodec	U2	n/a	n/a	n/a

To Do List (3/29-4/5)	Status
EE462 - Headphones	
Finish PCB!!!	
Split. .STL File!!	
Order Parts	
Update Schematic	
Organize Class Schedule!!	
Figure Out Workout Schedule	
Update LinkedIn	
Job Applications	
GameReady Replacement Parts	
Trip Planning	
Sign Up for Softball	
Diet I Guess	
n/a	

PCB Design Rev. 2.0

2x Audio Codec
1x BT Module
1x Speaker Amplifier
1x Audio Jack
1x Teensy

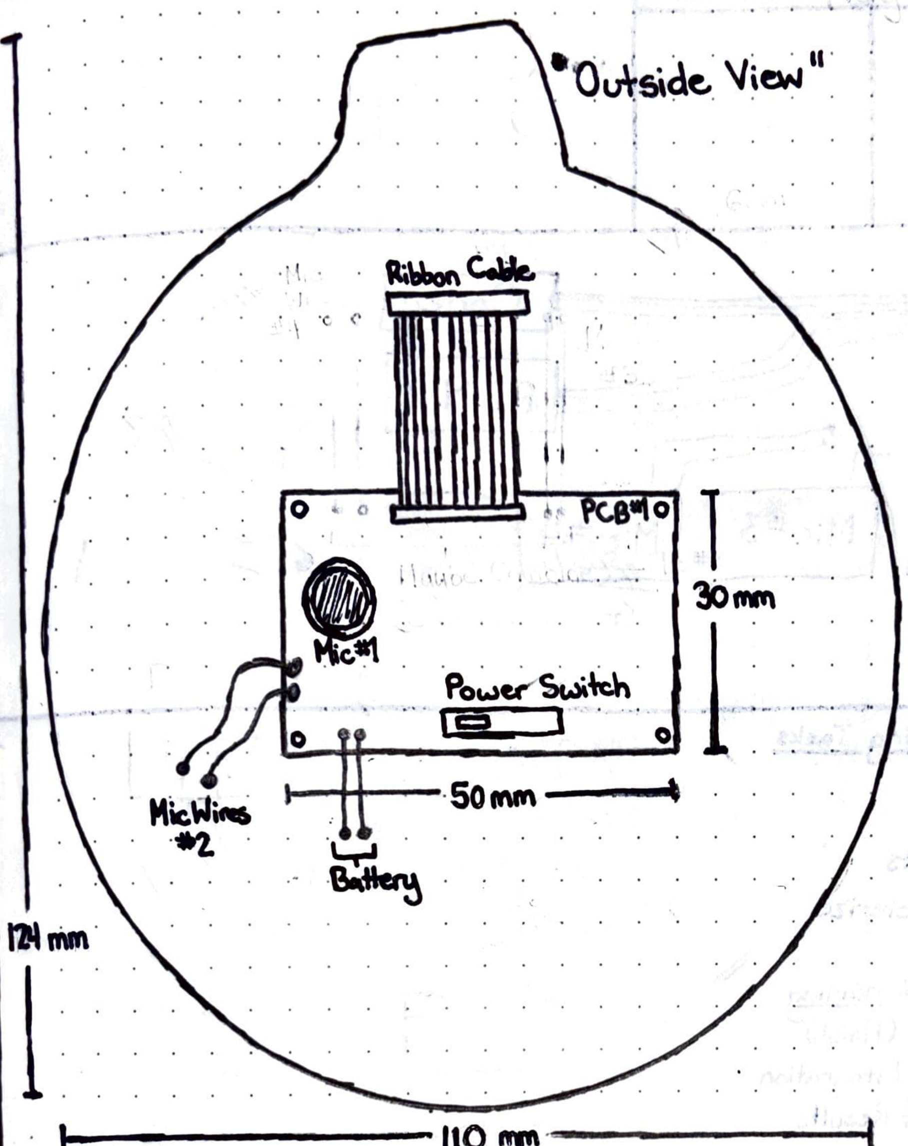
Preliminary Tech Specs | Headphone Left



By: Seth Cherry

Preliminary Tech Specs | Headphone Right

Circuit Components
4x Microphones
2x Speaker Drvrs
1x Spkr Amp
1x Audio Jack
1x MCU/DSP
1x BT Transceiver
2x Audio Codec



Headphone Right PCB	
Component	Quantity
Microphone	2x ✗
Audio Codec	1x ✗
Speaker Amp.	1x ✓
Power Switch	1x ✓
Volume Buttons	2x ✓

Headphone Left PCB	
Component	Quantity
Microphone	2x ✗
Audio Codec	1x ✗
MCU/DSP	1x ✓
BT Transceiver	1x ✓
Aux Jack	1x ✓

New PCB
- 4x Digital Mics
- 1x ADC for AUX
- Preamp?
- Anti-aliasing filter (Aux)
- Voltage (Buck & Boost)
- Headphone Amplifier
- ~~Volume Button Volume Buttons?~~
- Potentiometer
- Power Switch (3-Way)

By: Seth Cherry