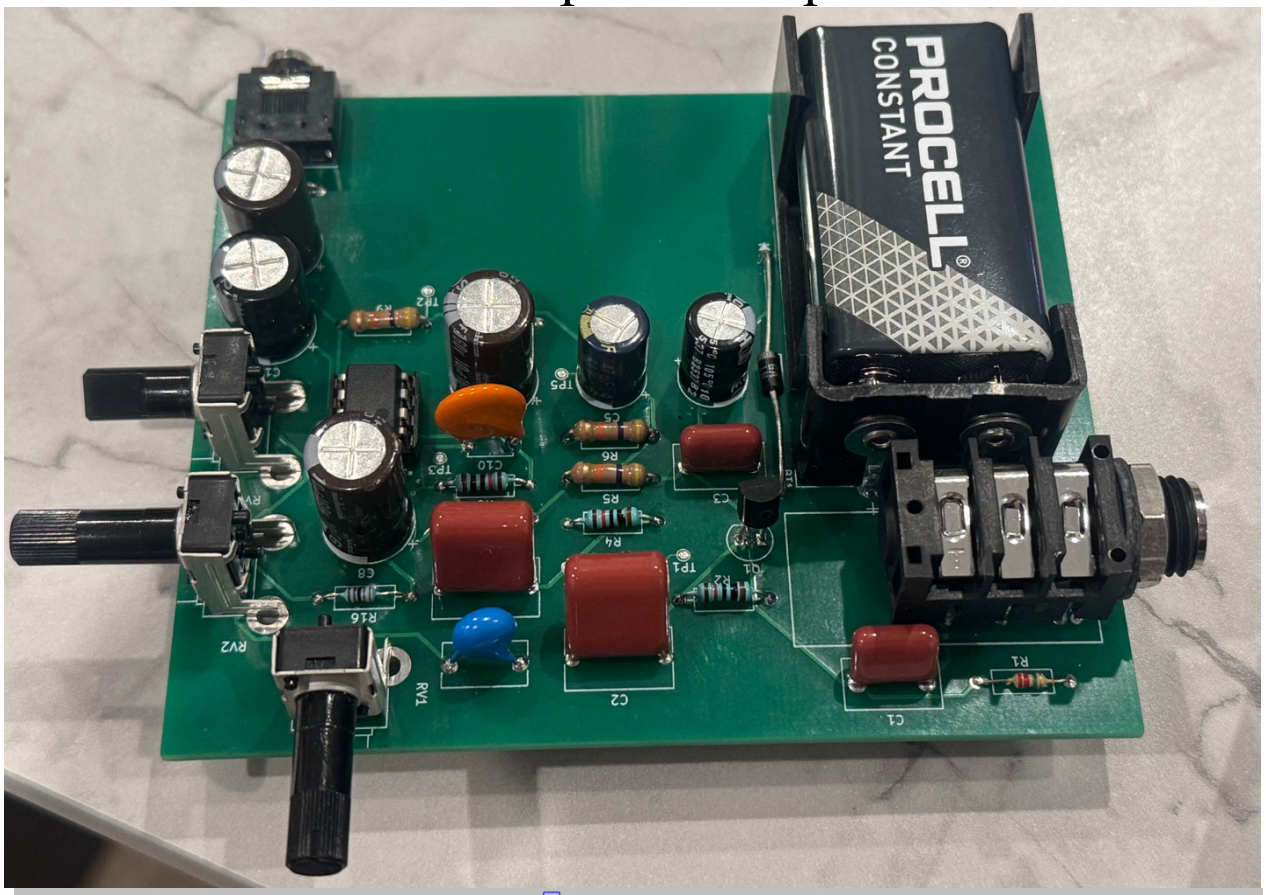
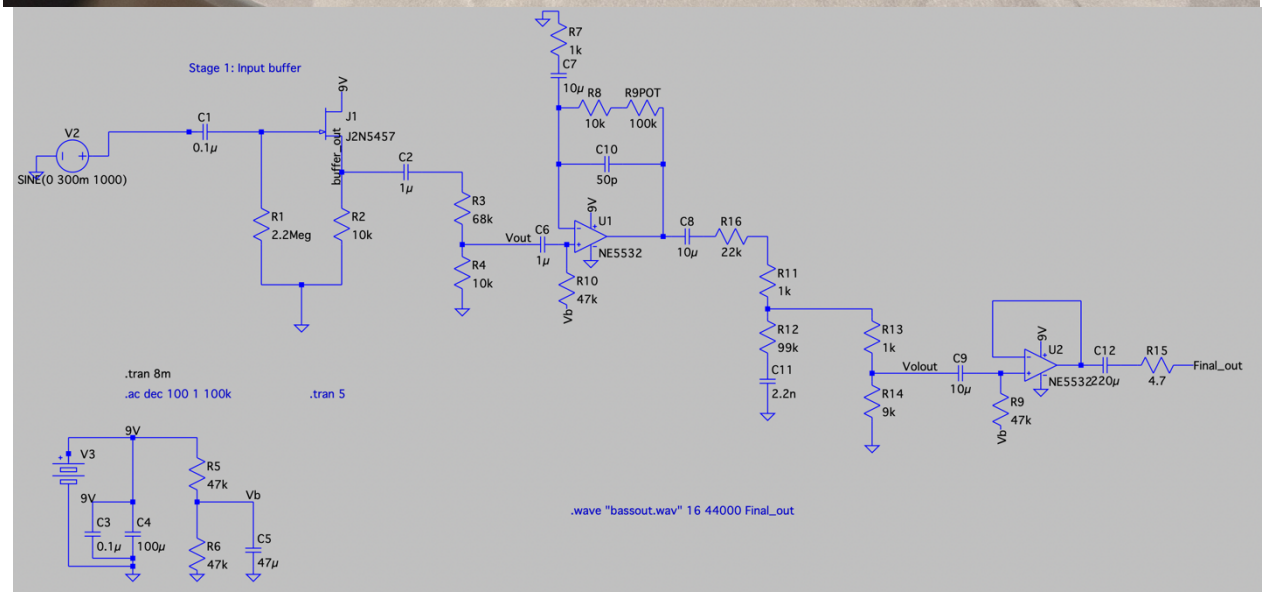
Bass Headphone Amplifier





AME 295

By Emma Wang

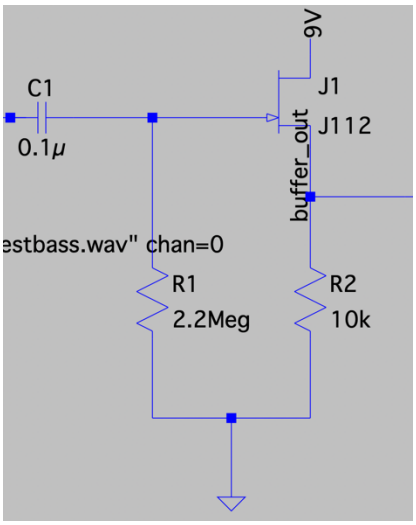
Theory of Operation

The bass headphone amplifier is a four-stage circuit, powered by a single 9V battery supply. It's designed for bass guitar player who plays an active bass and with to practice silently under some settings and not have to carry a big amplifier with them.

The signal chain starts from taking in the bass guitar input, going into a JFET input buffer. Then, the signal goes through a gain/overdrive stage, which is fed into a passive tone control (low pass filter) and then a volume control. The signal chain ends with an op-amp output buffer going into the headphone output. This circuit is designed for a Sadowsky active bass with approximately 300 to 500 mVpp input signal level driving 47 ohms headphones.

Power Supply

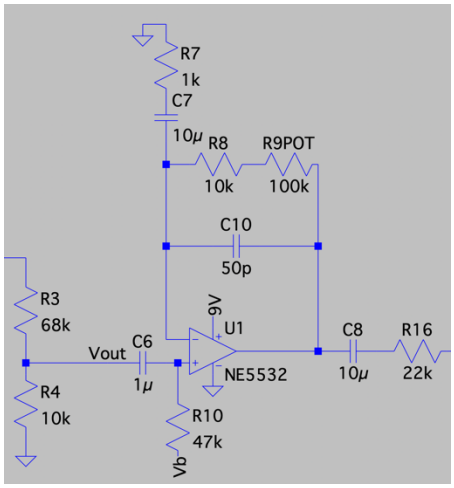
On an unbiased single supply, the rail would clip immediately on the negative cycle because the bounds are 0V and 9V, which means there's no negative headroom. To make the signal able to swing symmetrically with enough headroom, I created a virtual ground (Vb) at 4.5V using a voltage divider consisting of two 47k ohm resistors. Now, all op-amp input is biased to Vb through 47k resistors, and the signal is able to swing symmetrically around 4.5V with enough headroom on both sides. Some capacitors are used to filter noise and stabilize the bias voltage level.



Stage 1: JFET Input Buffer

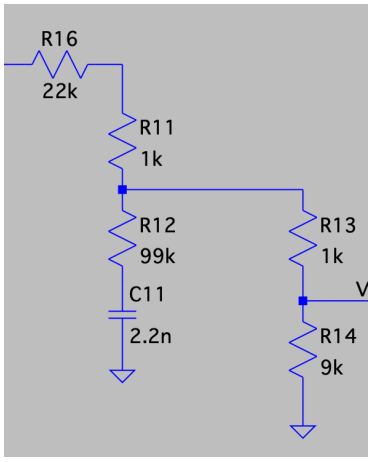
The first stage is a source follower using a J112 N-channel JFET. The original plan was to use 2N5457, but this series was discontinued. R1 sets a high input impedance going into the gate to avoid loading the pickup and lose signal contents. The source is connected to ground through R2 for self-biasing and the drain is connected directly to +9V. The purpose of input buffer is to transform the high source impedance to lower impedance that can be easily driven by the rest of the circuit. The gain should be almost at unity in this stage.

Stage 2: Gain/Overdrive



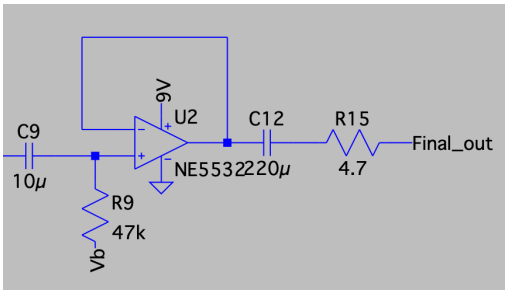
Before the second stage is a voltage divider that pads the gain by half to prevent clipping before gain stage. The gain stage using half of the dual op-amp NE5532 (U1A). This is a non-inverting configuration, in which the non-inverting input is biased to Vb. The feedback loop consists of a variable potentiometer and a fixed resistor R8 that connects the output to the inverting input. With the R7 connecting from inverting terminal to ground, the gain of the op-amp is established - with R9POT ranges from 0 ohm to 100k0hm, the gain ranges from 11x to 111x. C7 was used as AC coupling capacitor to block the DC bias, while C10 were used to roll off high frequencies above audible band to ensure stability. Driving the gain to higher range will overdrive the op-amp, which produces distortions through symmetrical clipping of the output.

Stage 3: Tone and Volume Controls



The gain stage output passes through R16 that limits the level again to approximately +/-600mV maximum at the headphone output by forming a voltage divider with the tone control resistors. The tone control is a passive low-pass filter. In the schematic on the right, R11 and R12 should be seen as one single pot, so do R13 and R14. Turning the POT towards C11 will result in a darker sound. The goal was to shape tone while preserving bass's low frequency fundamentals.

Stage 4: Output Buffer



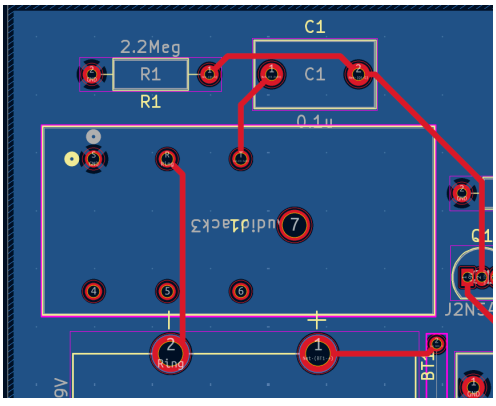
The output buffer uses the second half of the NE5532 (U1B). The op-amp is used as a voltage follower that produces unity gain. The buffer's purpose is to provide low output impedance capable of driving 47-ohm headphones. The R15 is used here to limit the peak current and again help guard the output volume to be within a safe range for hearing.

Power Switching

Instead of using a switch for power, I designed the input jack as part of the power control circuit. I connected the battery's negative terminal to the ring of the jack. When no cable is plugged in, since it's a stereo jack, the ring and sleeve are not connected, thus the circuit is open and no current flows. The amp is off in this state. Once an instrument cable is plugged in, the ring and sleeve of the audio jack gets bridged together, connecting the battery's negative terminal to ground. The circuit is powered as a result. I chose this technique to simplify the circuit and potentially the enclosure making process.

Troubleshooting Process

Input Audio Jack



The problem was identified in the earliest stage - when signal just enters from the input audio jack. We were able to see signal through the oscilloscope from one side of the pins but not the other side. This means that the signal was lost and didn't cross to the other side of the audio jack that's connected to the rest of the circuit. The problem was, there are six pins to the jack in 2 columns, and I couldn't get a certain answer from the spec sheet and ended up incorrectly assuming the TRS side. So, while I routed the TRS pin side to the rest of the circuit, the signal actually enters from the side of pin 4/5/6.

To resolve the problem, I used short wires to vertically bridge the upper and lower pins, just to short them and make the signal able to arrive to the other side and gets passes to the rest of the circuit.

JFET Pin Mapping

The original design used a 2N5457 N-channel JFET, which has been discontinued and became unavailable. I substituted it with a J112 N-channel JFET as it has similar electrical characteristics. However, during testing the signal was present at the gate but completely died at the source. This buffer doesn't produce any output.

The cause was simply a pinout mismatch between the two JFETs. The 2N5457 pin order is D-G-S, while the J112 pin order is D-S-G. In the design footprint, I used the 2N5457 footprint and ended up inserting J112 as it is without checking the pin order. As a result, the signal actually being fed into the source (the middle pin) of J12 instead of the gate.

The problem was fixed by carefully removing the component from the board and replace it with a new one. It was a good plan to get two of the same components, just in case it's hard to reuse a single component when the pins are already trimmed. The pins were extremely close and thin. It took me sometime to reconfigure the component without having pins touching each other which shorts the signal. It's always a good idea to recheck the pin out when component is substituted spontaneously.

Headroom for Clean Tone

A design limitation resulted from tradeoff of using a single 9V supply was identified during the simulation stage - the circuit is not able to produce a perfectly clean tone across all levels of playing. The minimum gain of the gain stage is 11x. Given that an active bass can output voltage levels ranging from 300-500 mVpp. At 11x gain, this gives 3.3-5.5 Vpp of output. However, I just found out that NE5532 is not a rail-to-rail amp, which means the op-amp on a single 9V supplied that's biased at 4.5V could only swing about +/-3V, instead of +/-4.5 fully to rail. As a

result, even at the minimum gain setting, the signal still clips slight if playing a little more aggressively. Since the tradeoff is fundamental, I would use an 18V supply to resolve this fully. This would double the available headroom to provide enough room for a completely clean signal.

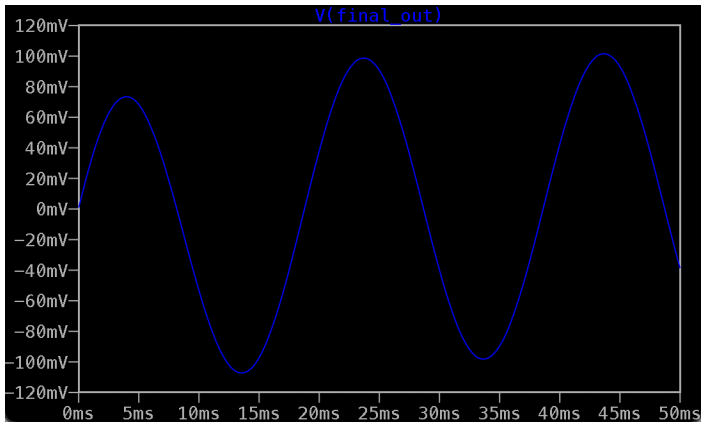
LTSpice Simulation Errors

I spent a lot of time building the NE5532 SPICE model for a more accurate simulation. Originally, I made LTSpice refer to the .lib file I created for NE5532 that's placed in the default library folder. It didn't take long to realize that when referencing externally, the default library folder path cannot be resolved. As a result, I embedded the entire definition directly in the schematic as a SPICE directive, so that LTSpice can source directly within the file.

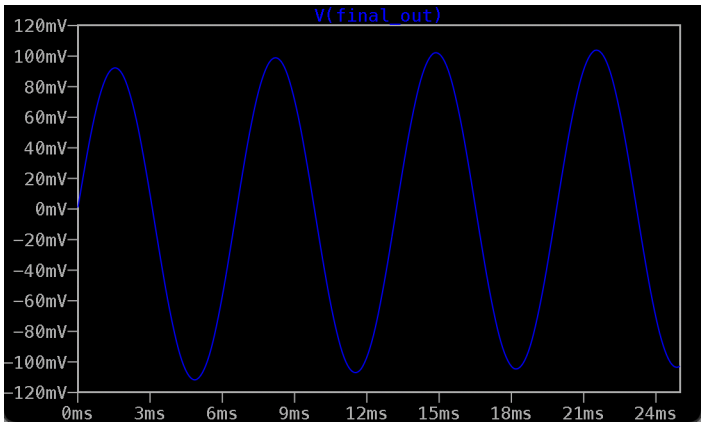
There were also times when a simple modeling error was thought as a circuit design error. For example, there's one time I incorrectly entered the unit suffix for megaohms. I wrote "2.2m" instead of "2.2Meg", which tricked LTSpice into thinking the unit is in milli and made a resistor a short to ground. It took significant troubleshooting time to find out about this error, even though it's straightforward.

LTSpice Simulation Result

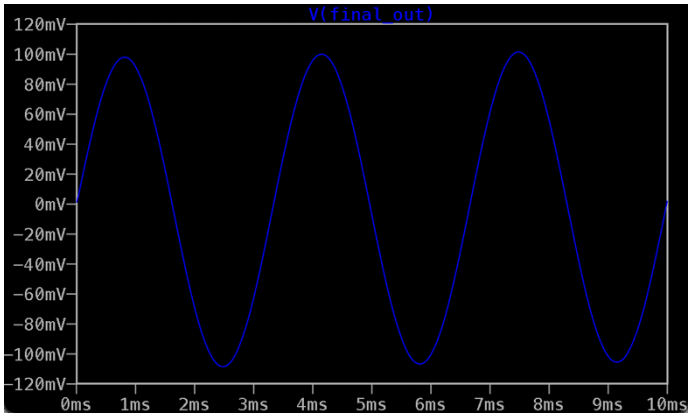
Sine wave 50mVpp at 50Hz



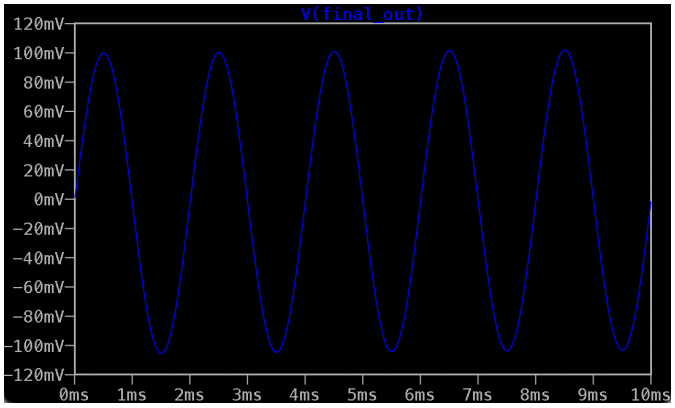
Sine wave 50mVpp at 150Hz



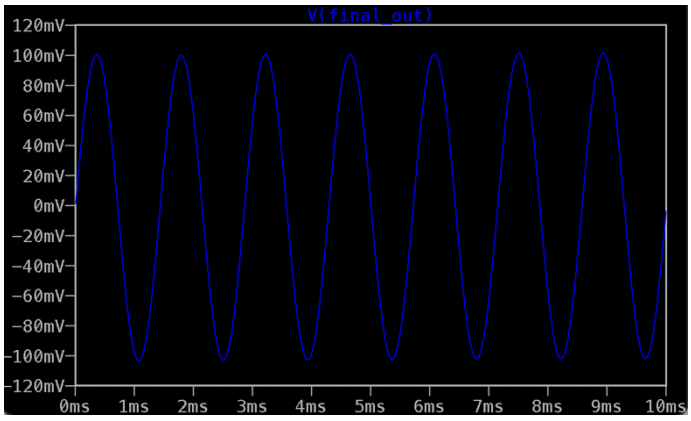
Sine wave 50mVpp at 300Hz



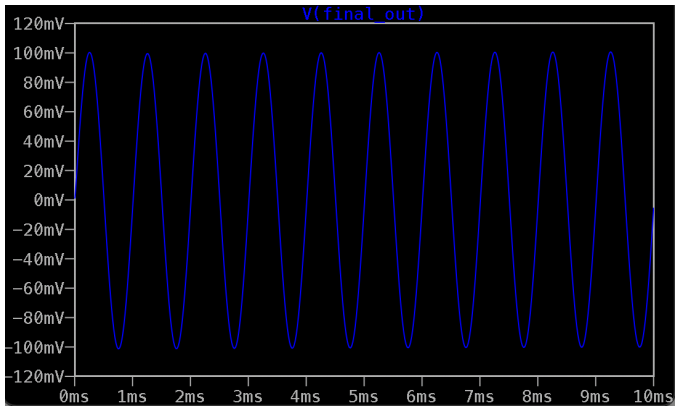
Sine wave 50mVpp at 500Hz



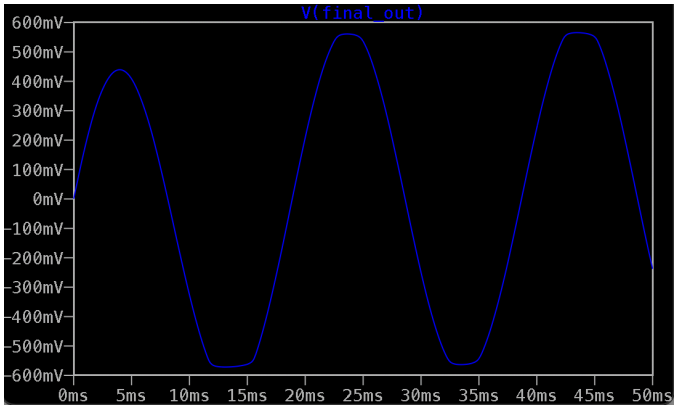
Sine wave 50mVpp at 700Hz



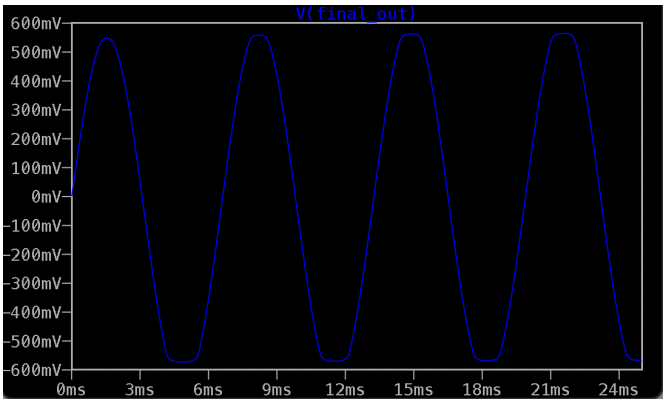
Sine wave 50mVpp at 1000Hz



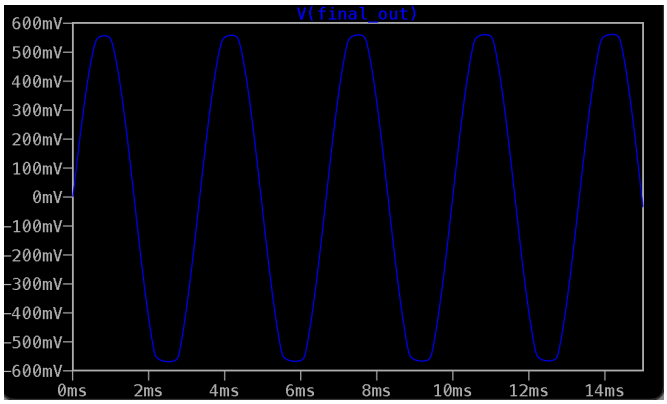
Sine wave 300mVpp at 50Hz



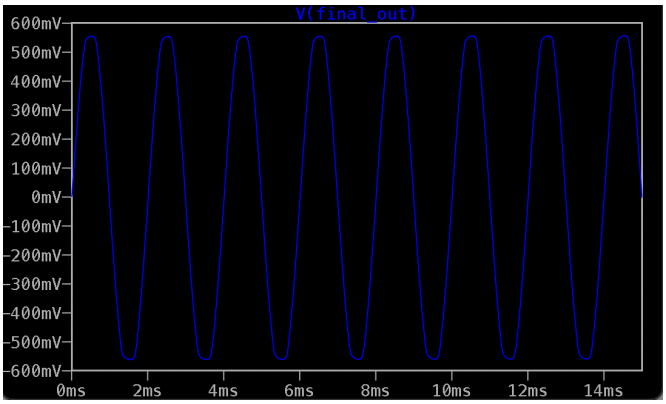
Sine wave 300mVpp at 150Hz



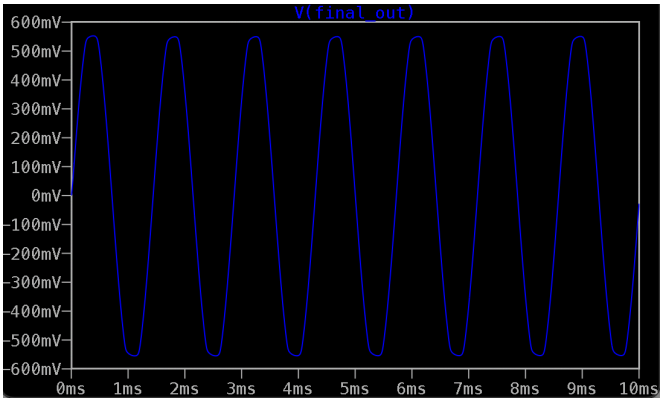
Sine wave 300mVpp at 300Hz



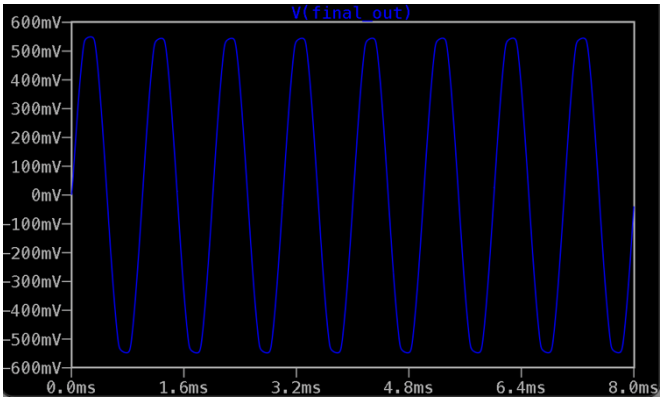
Sine wave 300mVpp at 500Hz



Sine wave 300mVpp at 700Hz

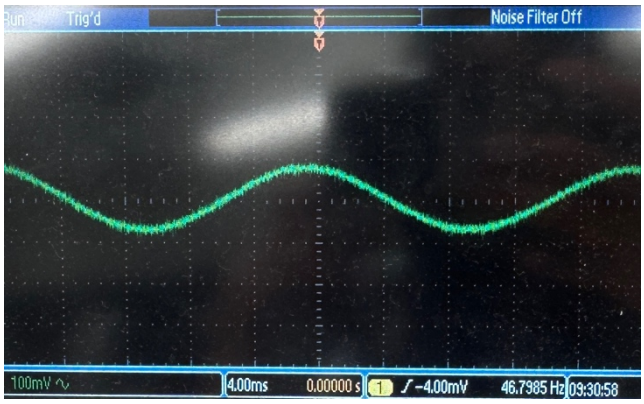


Sine wave 300mVpp at 1000Hz

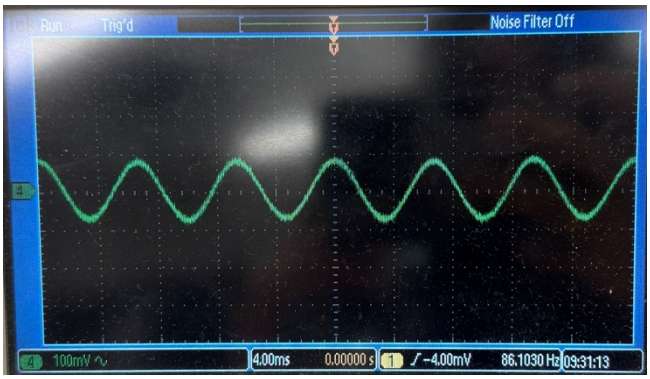


Actual Circuit Result

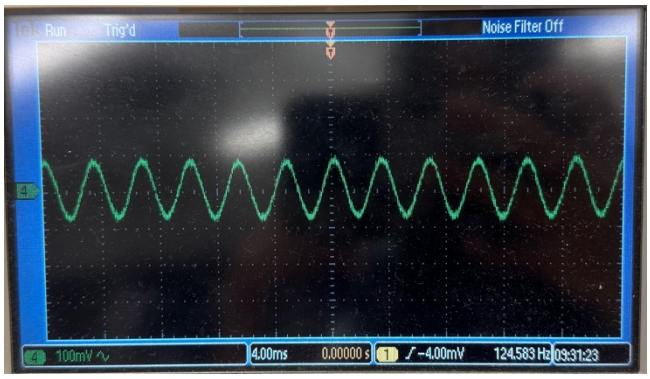
Sine wave 50mVpp at 50Hz



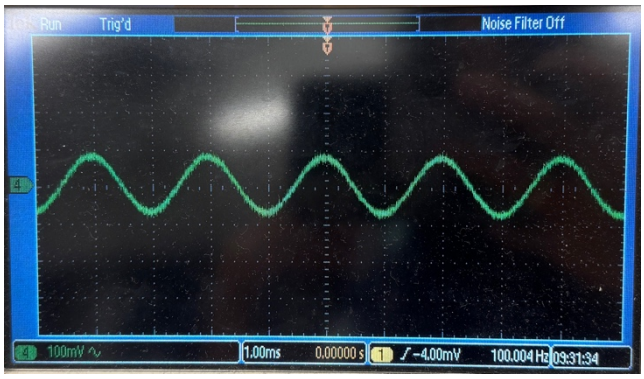
Sine wave 50mVpp at 150Hz



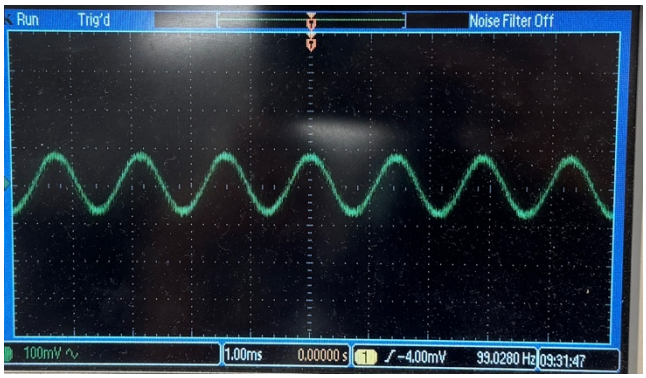
Sine wave 50mVpp at 300Hz



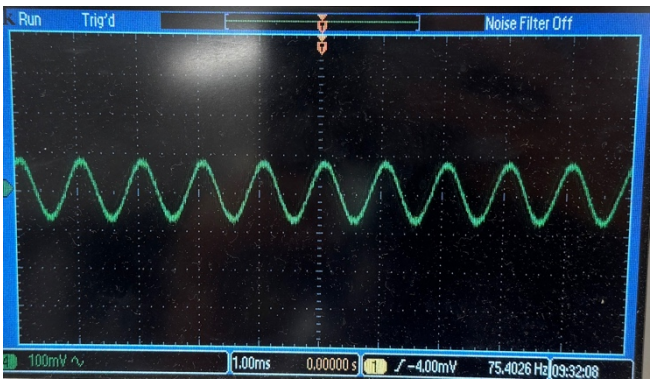
Sine wave 50mVpp at 500Hz



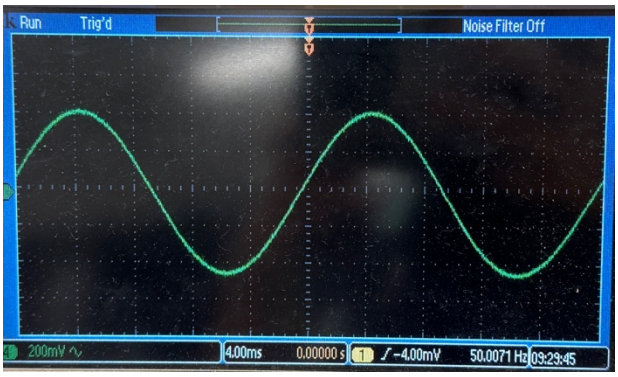
Sine wave 50mVpp at 700Hz



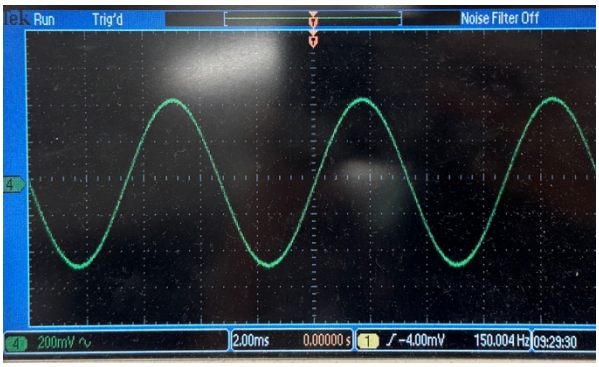
Sine wave 50mVpp at 1000Hz



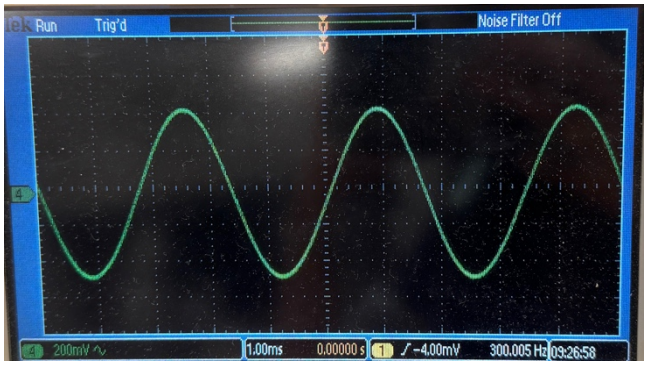
Sine wave 300mVpp at 50Hz



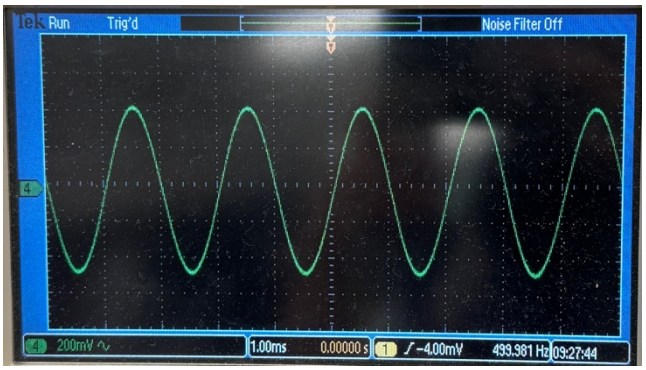
Sine wave 300mVpp at 150Hz



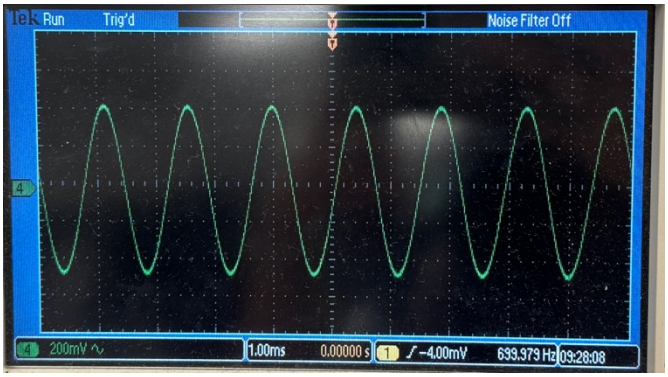
Sine wave 300mVpp at 300Hz



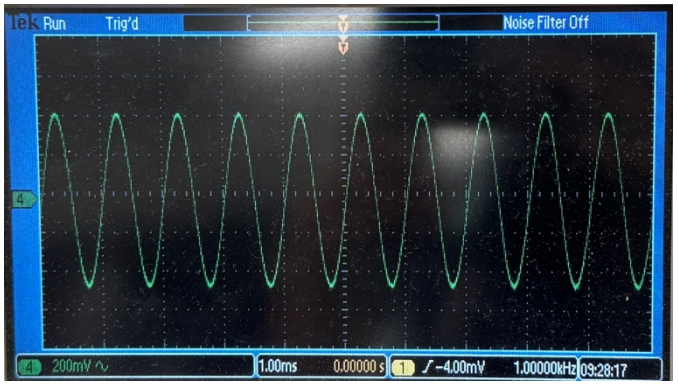
Sine wave 300mVpp at 500Hz



Sine wave 300mVpp at 700Hz



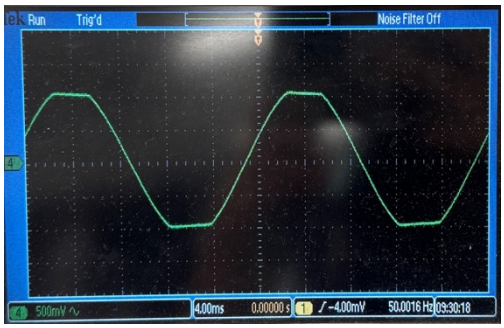
Sine wave 300mVpp at 1000Hz



Discussion

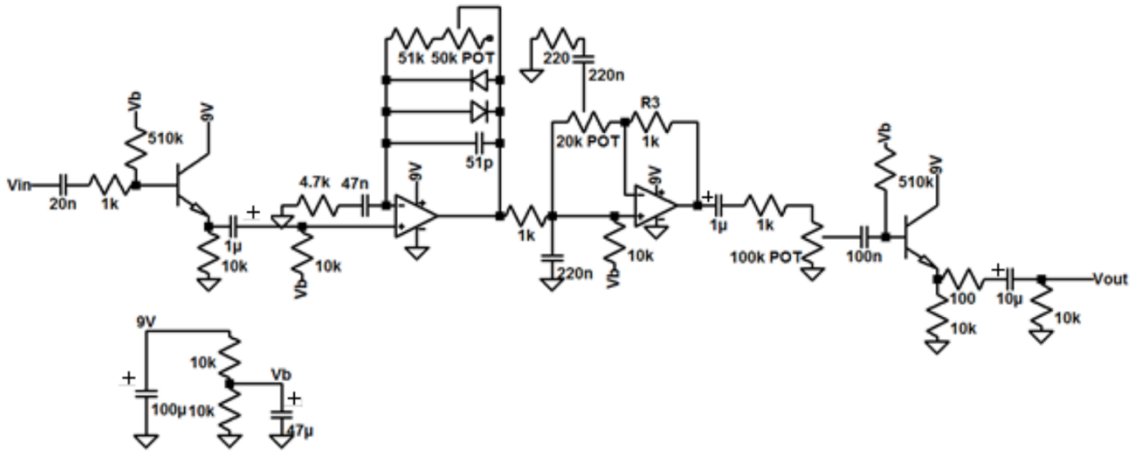
For both the simulation and actual testing stage, I fed sine wave as input and set all the knobs to their max, which gives max gain and drive and max volume. At 50ms, with small input signal, both the simulation and actual waveform are clean and smooth without any clipped peaks. Looking at the simulation result at 300mV input, signals are slightly clipped across all the frequency band tested, which suggests the presence of distortion. This aligns with the result when I plug in my bass and play - I could hear clear distortion within a wide range of playing dynamic.

What is odd is this distortion pattern didn't show up on the actual test with the oscilloscope. When looking at the sine wave at 300mV graphs above, all of them shows a clean sine wave. This means if my assumption about an 300mV to 500mV active bass input signal is correct, I shouldn't hear distortion like I did while playing at normal dynamic. To further investigate, I turned the input signal up to find where it starts to distort. Turns out, shown by the waveform on the right, the distortion happens when input level reaches 800mVpp.



Several guess towards the cause can be made. First, I underestimated the average input level of an active bass signal that is actually above 1Vpp. Second, the input buffer JFET substitute J112 may have a different tolerance, internal resistance, and buffer gain than 2N5457, which resulted in the actual circuit having a different gain characteristic.

Reference



The primary reference of this project is the Lab3 tube screamer circuit. It demonstrated very basic use of non-inverting op-amp for gain and overdrive to get a great distortion sound. I tried to adapt this[1] core concept and redesign it into fitting bass guitar playing by tunning the capacitors to different cut-off frequencies.

The article "How to design a basic distortion pedal circuit" is also referenced for the tone control approach, which is a POT with a capacitor to ground acting as a low-pass filter [1].

What I Considered of Value

The most valuable aspect of this final project is to go through a full cycle of circuit design from zero. From creating a LTSpice schematic to PCB design to getting a working physical circuit board, nothing was built off of any established templates of someone else's work. I tried to

[1] B. Wampler, “How to design a basic distortion pedal circuit,” Wampler Pedals, https://www.wamplerpedals.com/blog/lifestyle-hobby/2024/08/how-to-design-a-basic-distortion-pedal-circuit/ (accessed Apr. 11, 2026).

manifest with exactly what I learnt, which might turn out to be a little basic and minimal, but I was able to fully understand every step leading to the completion of this project.

More specifically, I spent a significant amount of time calculating all the coupling capacitors and tried to verify them through simulation first. For example, the tone control capacitor C11 was deliberately chosen to have a cutoff frequency of 723 Hz. I wanted it to be low enough to cut harsh clipping harmonics for a darker sound while still high enough to preserve bass fundamental frequencies. There are many other choices like this.

I also spent a lot of time selecting specific components and had some insights and deeper understanding towards component specs. For example, I learnt that there's an entire family of options for a 10uF capacitor that are not exactly the same thing. They span different voltage ratings, size, lead spacing, etc. Some of the capacitors I initially chose only by capacitance value had a voltage rating for 160V, which is a massive overkill for a 9V circuit, plus that the part is comparably larger. I ended up choosing much smaller voltage rated capacitors that are around 35V which also have a more appropriate size.

Translating circuits into KiCad footprint was also not easy work. I also had to consider the availability of footprint when selecting components. Even though I was being careful, there was still a mismatch between the input audio jack and the provided footprint. I was able to resolve it by trimming the pins to fit into the thinner holes.

The debugging process is also very worth-mentioning. The circuit did not work on when powered up for the first time, which should be expected. I learnt the value of understanding signal flow and using oscilloscope as a tool to do signal tracing for troubleshooting. Specific problems were mentioned in the challenge section, but I learnt to quickly pin point problem by identifying the dead points of signal and then work on eliminating causes.

From this project, I also understood the gap between simulation and reality. The value in simulation doesn't lie in perfectly simulate the reality, but the approximation was close enough to get me a working hardware product on my first design attempt.

Back of Circuit Board