

# BME 194: Applied Circuits Lab 9: class-D power amplifier

Kevin Karplus

February 27, 2013

## 1 Design Goal

In lab 5, you made an audio amplifier for a microphone using an op amp, but it was not powerful enough to provide loud sounds from a loudspeaker, because the op amp had limited output current, and the loudspeaker had a low impedance so the power to the loudspeaker,  $I^2R$ , was small. (With a 23mA limit and  $8\Omega$  the maximum power is about 4mW.)

In this lab, you will design an amplifier that is capable of delivering 10W of power to loudspeaker, over 2000 times more than the one-op-amp audio amplifier of lab 5.

Because you will be using power field-effect transistors (FETs), a class-D amplifier design is probably the easiest to get working. [5]

Design constraints: In addition to the parts in your kit, you should use the triangle wave output from the function generator and both the 6v power supply and the  $\pm 25V$  dual power supply. The loudspeaker should be powered from the dual supply (though not at 25V!), but the preamplifier from the 6V supply.

## 2 Background

### 2.1 Real power

To compute the energy dissipated in an impedance you need to multiply the instantaneous voltage by the current, then integrate over time. To get the average power, you divide the energy by the time you integrated over.

For example, if the voltage across a device is  $V(t) = A \cos(\omega t + \phi)$  and the current into the device is  $I(t) = B \cos(\omega t + \psi)$ , then their product is

$$AB \cos(\omega t + \phi) \cos(\omega t + \psi)$$

or, using the trig identity  $\cos(a) \cos(b) = 0.5(\cos(a + b) + \cos(a - b))$ ,

$$0.5AB (\cos(2\omega t + \phi + \psi) + \cos(\phi - \psi)) .$$

Note that there is a periodic part, with frequency twice as high as the waveform we started with and a constant part. If you average the power over a full period, the periodic part integrates to 0, and you have just the constant part:  $0.5AB \cos(\phi - \psi)$ .

Note: for a resistive load, current and voltage are in phase ( $\phi = \psi$ ), the cosine of the difference is 1, and the real power is  $0.5AB$ , just as we computed when we talked about RMS voltage earlier this quarter.

It turns out to be easier to analyze sinusoids using *RMS phasors*, representing

$$V(t) = A \cos(\omega t + \phi) \text{ as } \bar{V} = Ae^{j\phi}/\sqrt{2} ,$$

and

$$I(t) = B \cos(\omega t + \psi) \text{ as } \bar{I} = Be^{j\psi}/\sqrt{2} .$$

So a voltage amplitude of  $A$  results in an RMS voltage of  $A/\sqrt{2}$  and a current amplitude of  $B$  results in an RMS current of  $B/\sqrt{2}$ . The main point of RMS voltage and RMS current is that you can multiply them to

compute power as  $\bar{V}\bar{I}^*$ , where the asterisk represents the complex conjugate (negating the imaginary part, which has the same effect as negating the phase). This gives us the right result for voltage and current being in phase ( $\phi = \psi$ ), no matter what phase that is.

If  $\bar{V}$  and  $\bar{I}$  are RMS phasors, then  $\bar{V}\bar{I}^*$  is called the *complex power*,  $|\bar{V}\bar{I}^*|$  is called the *apparent power*,  $\text{real}(\bar{V}\bar{I}^*)$  is called the *real power*, and  $\text{imag}(\bar{V}\bar{I}^*)$  is called the *reactive power*. Power engineers like to change the names of the units (watts for real power, volt-amps for apparent power, and vars for reactive power). They're all the same unit, but changing the name reminds you that they are not computed the same way and are not interchangeable. You will not be held responsible for those names in this course, and may use watts for all forms of power, but we will expect you to mean “real power” unless you explicitly modify the noun.

The real power is what you computed by doing the integral of the cosine waves—it represents the power actually dissipated in the device. Note that a sine wave generator dissipates negative power (the current into the device and the voltage have opposite signs).

The reactive power represents power that is being temporarily stored in the device during part of the cycle, and released in a different part of the cycle. If you have a purely reactive circuit (just capacitors and inductors with no resistors, so the impedance is purely imaginary), then the voltage and current are  $90^\circ$  out of phase, and the real power is  $0.5AB \cos(\pi/2) = 0$ . No energy is dissipated in a purely reactive circuit.

Note that if you use *amplitude phasors* (representing  $A \cos(\omega t + \phi)$  as  $\bar{V} = Ae^{j\phi}$  without the  $\sqrt{2}$  correction) rather than *RMS phasors*, the power computation  $\text{real}(\bar{V}\bar{I}^*)$  gives us double the actual power. That is why our loudspeakers are rated as 10W RMS power (the actual power they can dissipate safely) or 20W peak power (the number you get if you multiply the amplitude phasors).

If you know the RMS voltage phasor  $\bar{V}$  for a sinusoid to a complex impedance  $Z$ , the RMS current phasor is just  $\bar{V}/Z$ , and the real power is  $\bar{V}(\bar{V}/Z)^*$ .

You'll want to compute the real power that your amplifier delivers to the loudspeaker, to make sure you don't exceed 10W RMS. You'll also want to compute how much power is delivered at frequencies that do not produce useful sound (like the PWM frequency discussed in Section 2.5), to make sure that you are not wasting power heating up the loudspeaker to no useful effect.

Manually computing the complex impedance of a loudspeaker or a complicated network of resistors, capacitors, and inductors is tedious and unnecessary. I will provide models for the loudspeaker that you can use with gnuplot to plot the impedance or power (or use the “print” statement to do a calculation of a single value).

## 2.2 Setting the power supply

If you want to deliver 10W (the maximum RMS power) to your  $8\Omega$  loudspeaker, what is the maximum voltage and current you should apply? You should look at most powerful signals you can deliver to the speaker, which would be square waves that alternated between the positive and negative voltage rails. If you are using the symmetric bench supply for driving the loudspeaker, with one end of the loudspeaker on the 0V ground in the middle, the maximum power you can deliver is  $(\pm V)^2/R$  or  $I^2R$ , so your power-supply voltage should be limited to about 9V, and the current limited to 1.1A, assuming an  $8\Omega$  resistive load. Your speakers may be closer to  $7.5\Omega$ , which would mean limits of 8.7V and 1.15A.

Note that the MCP6004 op-amp chips you are using cannot be powered from a 9V power supply, so you should use a separate 6V supply to power any op-amp chips you use.

## 2.3 FETs as output stage

In this class, you use FETs mainly as non-linear, digital devices. For example, in lab 6, you turned an FET on or off to provide square wave input to a loudspeaker. One reason for using the power FETs as switches is that they are very energy efficient if used that way (they dissipate little power as heat). Another reason is that their characteristics are very temperature-dependent, if used as linear devices, so designing linear circuits with power FETs is quite difficult (beyond the scope of this class).

Why are FETs power efficient as switches?

When an FET is off, it has a very high resistance between the source and drain and passes little current. I measured about 1nA for the NTD5867NL nFET transistor with  $V_{GS} = 0V$  and around 100nA with  $V_{GS} = 1V$ , though its data sheet allows up to  $100\mu A$  with for  $V_{GS} = 0V$  and  $V_{DS} = 60V$ , if the transistor warms up to  $125^\circ C$ . Note that even with the data sheet worst-case conditions, the transistor would only dissipate  $60V \cdot 100\mu A = 6mW$  of heat when it is off. If you use it with  $V_{DS} < 10V$  and keep the transistor from heating up, it should dissipate less than  $10\mu W$  when it is off.

When an FET is on, it has a very low resistance, and so the voltage drop across it is small. The data sheet for the NTD5867NL nFET specified an on-resistance of at most  $50m\Omega$  when  $V_{GS} = 4.5V$  and  $39m\Omega$  when  $V_{GS} = 10V$ . If you use the FET to drive an  $8\Omega$  loudspeaker, the current will be limited by the power supply driving the speaker (a 9V power supply would be delivering 1.125A), so the power dissipated in the on FET would be  $I^2 R = 63mW$ , while the power to the loudspeaker would be about 10W. The transistors are designed to handle large currents (up to 20A if kept cool, 13A if allowed to get up to  $100^\circ C$ ), which are much larger currents than you'll be dealing with). The main limitations of the FETs when they are fully on, though, is with heating: if you try to put 20A through them they'll dissipate  $(20A)^2 50m\Omega = 20W$  and quickly heat up past their maximum temperatures. Incidentally, it is easy to get these FETs hot enough to melt the plastic on the breadboards—I have done so.

Although the nFETs you are using are supposedly capable of dissipating 36W of power without damage if kept at  $25^\circ C$ , you don't want to go there, as you can't keep them that cool. The junction-to-ambient thermal resistance with a heat sink is  $45^\circ C/W$ , so keeping the junction below the  $150^\circ C$  maximum junction temperature limits us to about 2.8W dissipated by the nFET. Without a heat sink, as you are using them, you need to keep the power dissipation in the FETs even lower.

When FETs are switching between the off and on states they can have intermediate resistance values and dissipate large amounts of power. For example, if you were driving an  $8\Omega$  load and used an FET to get the voltage only half way to the power rail, then the FET would have the same resistance as the load, and each would be dissipating the same amount of power (2.5W in the example of a 9V power supply). To prevent overheating the FETs, try to switch between the on and off states as quickly as possible.

The standard configuration for an FET output stage in an amplifier (either a class-D amplifier like you're designing or in a CMOS digital circuit) is to have one nFET and one pFET, with their drains connected to the output and their sources connected to the power rails, as shown in Figure 1. The FET output stage works great if one of the two FETs is turned on and the other off, providing a low-resistance connection to the desired voltage rail. If both FETs are turned on, though, it can pass a large current from one power rail to the other, heating up the FETs to no useful effect.

Various circuits can be used to ensure that you don't have both FETs on at once. You'll be using the simplest of them: just connecting the two gates together and switching the gate voltage very rapidly from a voltage near the top power rail to a voltage near the bottom power rail. Note that when the gate voltages are near the top power rail, the pFET has  $V_{GS} = 0V$  and the nFET has  $V_{GS} = 18V$  so the pFET is off and the nFET is on. When the gate voltages are at the bottom power rail, the pFET has  $V_{GS} = -18V$  and the nFET has  $V_{GS} = 0V$ , so the pFET is on and the nFET is off.

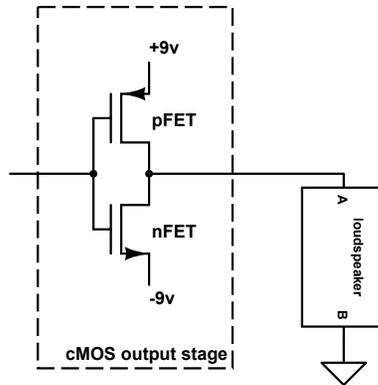


Figure 1: A cMOS output stage, consisting of one pFET with its source connected to the positive voltage rail and one nFET with its source connected to the negative voltage rail, and the output from the two connected drains. Here pin A of the loudspeaker is connected directly to the drains of the FETs, but see Section 2.7 for information about adding an output filter.

## 2.4 LM2903 comparators

Your cMOS output stage needs to have gate voltages that go rapidly from the lower power rail to the upper power rail, but our MCP6004 op amps are not capable of doing that. Not only are they limited to a maximum voltage swing of about 6V, they have a designed-in speed limitation: a slew rate of  $0.6\text{V}/\mu\text{s}$ , so it would take them  $10\mu\text{s}$  to do even their full 6V swing.

Because of these limitations of the op amp, I've provided another chip in your parts kit: the LM2903 dual comparator chip. The comparator chip can have a 30V difference between the top voltage and bottom voltage, so there is no problem powering it from your dual-rail supply, which has only an 18V difference.

The LM2903 has a different sort of output than the op-amps and Schmitt triggers you've used so far—an open-collector output. When the positive input is higher than the negative input, there is essentially no current through the output, but when the positive input is lower than the negative input, the output can sink up to about 7.5mA. If you look at the data sheet, you see a couple of different numbers  $V_{OL} < 400\text{mV}$  at  $I_{OL} = 4\text{mA}$  and  $I_{OL} > 6\text{mA}$  at  $V_{OL} = 1.5\text{V}$ . You can sink about twice as much current by putting both comparators of the LM2903 in parallel (connecting corresponding inputs and outputs for the two comparators).

To get the output to ever go positive, you need a pull-up resistor to provide current when the LM2903 output is off and pull the output to the top power rail. Sizing that pull-up is a design task. Let's look at the constraints on it. The LM2903 and pull-up resistor need to drive the gates of the FETs high and low (enough to turn the FETs fully off) rapidly. The gates of FETs are essentially capacitors (you can look up the input capacitance on the data sheets for the FETs).

If you make the pull-up resistor very large, then a low output will be discharging the FET gates rapidly with about 10mA of current through the LM2903 outputs, but a high output will charge the FET gates very slowly through the large pull-up resistor.

If you make the pull-up resistor very small, then a low output will be looking sort of like a voltage divider, and you might not be able to get a low enough output voltage to turn off the nFET fully. That would result in a lot of wasted power through the cMOS output stage. You'd also have a few milliamps of current running through the LM2903 output stage and the pull-up, which might also be wasted power.

The optimal design would be a small enough value to keep the RC time constant small for a high output and yet large enough to ensure that the low output shuts off the nFET. It probably suffices to design the output to be at about 1V above the negative power rail when the output is low (estimate the max current

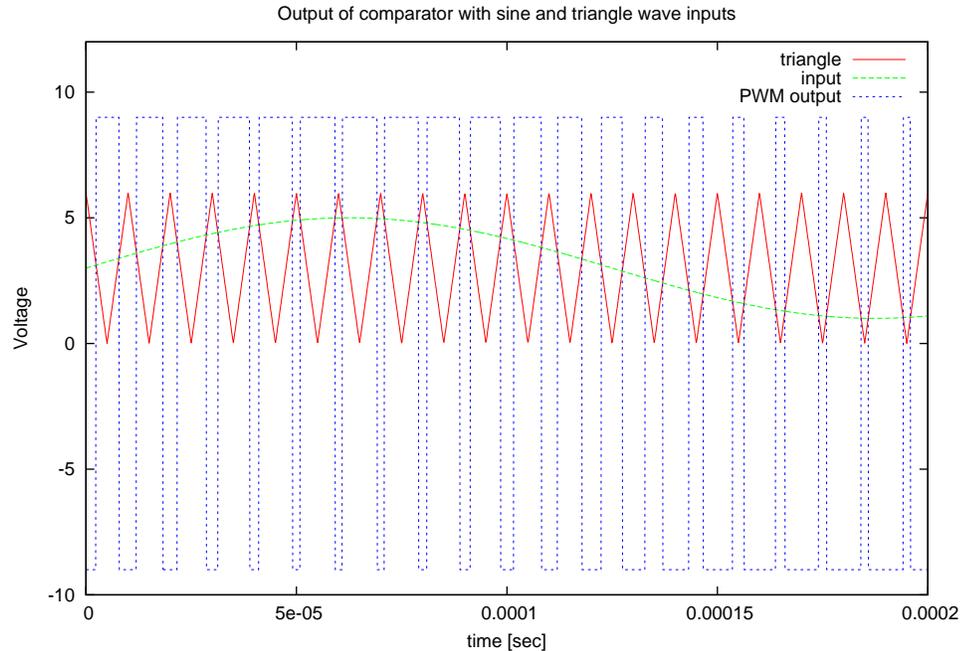


Figure 2: The output of a comparator for different input voltages with a triangle wave on the negative input of the comparator. The comparator power inputs are set to +9V and -9V. Note that the DC offset and amplitude of the triangle wave affect the behavior of the comparator. Here I've used a  $\pm 3V$  triangle wave centered at 3V.

as about 7–8mA per comparator, or 15mA for the two in parallel).

There is likely to be some asymmetry between how fast you can shut off the nFET and how fast you can shut off the pFET—you'd like both to be as fast as you can reasonably make them. y

## 2.5 Pulse-width modulation (PWM)

The idea behind class D amplifiers is that you don't provide the output as a voltage that is a fixed multiple of your input voltage (the classic idea of a linear amplifier). Instead, you provide a pulse train that rapidly switches between the positive power rail (+9V) and the negative power rail (-9V). What you adjust is how much time you spend at the positive rail and how much time you spend at the negative rail. This is referred to as *pulse-width modulation*, abbreviated PWM, because you are adjusting the duration of the high pulses. [6] A PWM signal is usually specified with two parameters: its *frequency* and its *duty cycle* (what percentage of the period the output is high).

You can get a pulse-width modulated signal from a voltage input by using a comparator and a triangle wave generator. If the voltage of your input is higher than the triangle wave, then your output is high. If the voltage of your input is lower than the triangle wave, then your output is low. (See Figure 2.) The amount of time the output spends in the high state corresponds to how much of the time the triangle wave is lower than your input. Since the triangle wave spends an equal amount of time at each voltage, the fraction of the time spent high (the duty cycle) is proportional to the input voltage, and the frequency of the PWM signal is the same as the frequency of the triangle wave.

## 2.6 Inductive loads for PWM

Pulse-width modulation is often used for driving inductive loads like motors and loudspeakers. Both motors and loudspeakers are more dependent on the current through their coils than on the voltage across them, because the magnetic field strength of a coil is proportional to the current.

PWM works well with motors and loudspeakers because the rapid changes in voltage result in much slower changes in current. One gets sawtooth-like fluctuations in the current around the average value, and the average value is determined by the duty cycle of the PWM signal.

Furthermore, if the PWM signal has a very high frequency, the mass of the loudspeaker or motor keeps it from moving much in response to the high-frequency part of the waveform, so you only observe the sound waves or motor movement due to the lower frequency components.

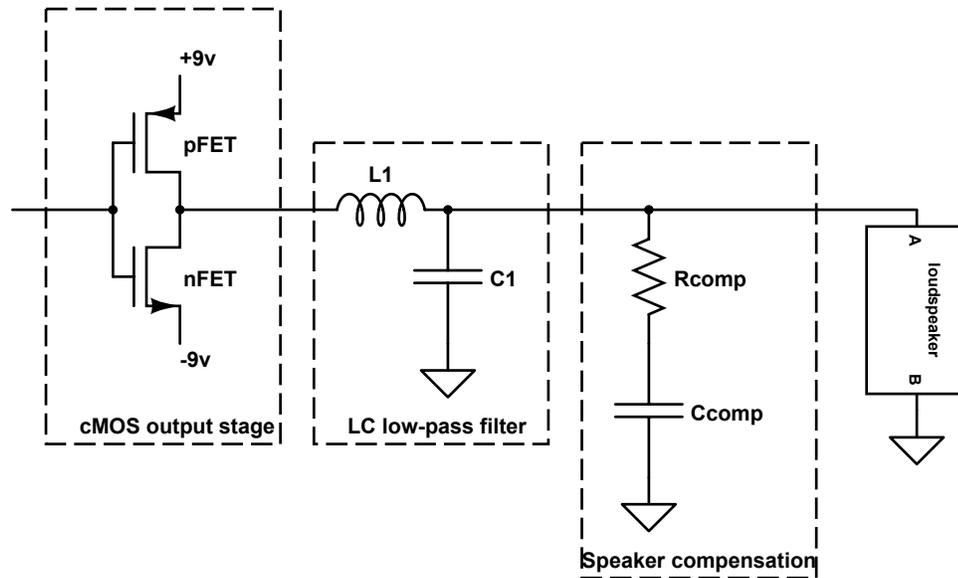


Figure 3: After the CMOS output stage of the class-D amplifier, an LC low-pass filter is usually added to avoid radio interference and clean up the current ripple inherent in class D amplifiers. To make the LC filter easier to design, there is usually an RC compensation network added to the loudspeaker, so that it presents a purely resistive load to the LC filter at high frequencies. See Section 2.7.1 for more information about the compensation, and Section 2.7.2 for more information about the LC filter.

## 2.7 Output filter overview

In professional class-D amplifiers for audio work, the output of the CMOS stage is filtered by an LC low-pass filter (see Figure 3) [3]. The corner frequency of the filter needs to be between the highest input frequency to be amplified and the frequency of the triangle wave, which is a much higher frequency.

Motor drivers often don't bother with the LC filters, though they can reduce heating in the motor and make electromagnetic interference easier to control. One reason is that motors require a lot of current, and power inductors capable of handling large currents are expensive. The little ones I bought for the class are only about 60 cents, but they have a saturation current of 1.9A—plenty for our 10W loudspeakers but not a lot for a motor.

You could skip the output filter in your design, and connect your loudspeaker directly from the drains of the pFET and nFET to the ground of the dual power supply. The amplifier would work, but you would have high-frequency ripple in the current to the loudspeaker, some of which may cause radio-frequency interference with AM radios.

### 2.7.1 Zobel network for speaker compensation

Professional designs often add a series RC circuit in parallel with the loudspeaker (called a “Zobel network” [4]) to compensate for the inductance of the loudspeaker and make it look like a purely resistive load at high frequencies. That compensation network makes the LC circuit easier to design, and the amplifier does not need to be designed for a specific loudspeaker—all compensated  $8\Omega$  loudspeakers will have the same electrical behavior at high frequencies.

This section will walk you through how to use gnuplot to design a Zobel network, and then why you probably don't want one for this lab. If you skip reading this section, it probably won't hurt your ability to design your power amplifier, but there are some good ideas here that are worth reading about, even

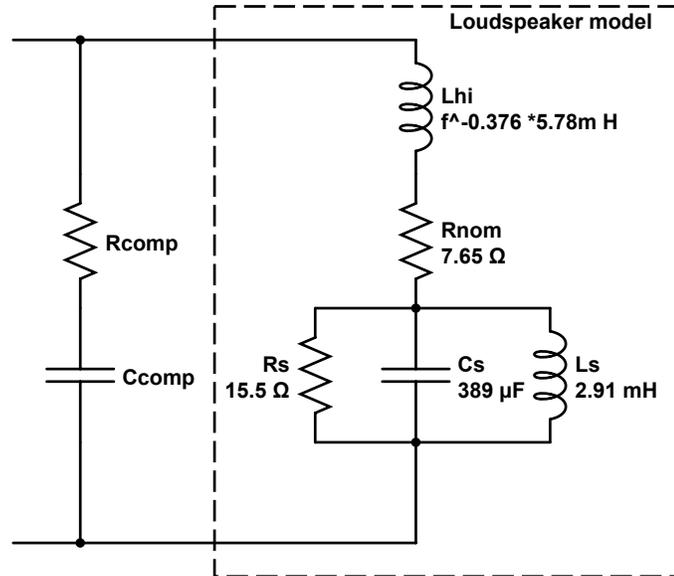


Figure 4: The box contains one electrical model for the JAMO 30466/30462 loudspeakers, based on voltage and current RMS measurements made by Kevin Karplus [2, 1]. The extra resistor  $R_{comp}$  and capacitor  $C_{comp}$  are added externally to make the overall circuit have nearly constant impedance above 400Hz. (They do nothing to erase the mass-and-spring resonance around 148Hz.)

if you don't incorporate them in your design.

To choose values for the resistor and capacitor, we need a model for the impedance of the loudspeaker. I have provided such a model in Figure 4. It is very similar to the model we derived in class, but uses an inductor whose inductance decreases with frequency, following a power law.

To choose values for  $R_{comp}$  and  $C_{comp}$ , gnuplot can be used to fit the parameters to get a nearly constant value above 1kHz, see Figure 5.

I tried doing the fits using both the power-law model of the loudspeaker that is shown in Figure 4 and the simpler model that we developed in class, using scripts like the one in Figure 5. They produced somewhat different values for  $C_{comp}$  and the value suggested by the more complicated model (see Figure 6 did indeed provide a more constant impedance than the value suggested by the simpler model when I tried them in the real world, though I did not make careful measurements.

**Warning:** At 100kHz, the added resistor has to dissipate 10W, which far exceeds the capabilities of our little 1/4-watt resistors (see Figure 7). Adding an LC low-pass filter limits the power to the speaker at the PWM frequency, but it is not enough to save us here, since even at 10kHz (within the range of frequencies you're likely to be interested in amplifying), the compensation network would need to dissipate over 7W.

Rather than buying expensive 10W resistors (OK, they're only about 35 cents each in the quantities we would need for the class) and have to worry about the heat from them and the reduced efficiency of the amplifier from the power they dissipate, I propose that you not use a Zobel network. The main advantage of the Zobel network is that it simplifies the design of the LC low-pass output filter and makes it independent of the choice of loudspeaker, but you can design your LC filter for your specific loudspeaker instead.

```

set logscale xy
set xlabel "frequency [Hz]"
set xrange[10:1e6]
set samples 1000
set ylabel "impedance [ohms]"
set yrange[5:500]
set title "Compensating JAMO 30466/30462 loudspeaker with Zobel network"
set key top left

j=sqrt(-1)
conjugate(a) = real(a) - j*imag(a)
phase(v) = imag(log(v))
real_power(V,Z)= real(V*conjugate(V/Z)) # V is RMS voltage phasor

Zc(f,C) = 1/(j*2*pi*f*C) # f= frequency [Hz], C=cap [farads]
Zl(f,L) = j*2*pi*f*L
Zpar(z1,z2) = z1*z2/(z1+z2)
divider(Zup,Zdown) = Zdown/(Zup+Zdown)

Rnom=7.64;
Lhi= 5.88e-3;  Lhipow = -0.378
Ls=2.92e-3;   Cs=386e-6;   Rs=15.5

#impedance of loudspeaker
Zloud(f) = Rnom+Zl(f,Lhi*f**Lhipow)+Zpar(Rs,Zpar(Zl(f,Ls),Zc(f,Cs)))

# impedance of whole circuit with RC compensator
Zcomp(f,Rcomp,Ccomp) = Zpar(Rcomp+Zc(f,Ccomp), Zloud(f))

#Guess some values
Rcomp=Rnom
Ccomp=0.1e-6

# Kluge: we need a points at which to do the fitting.
# Instead of creating a data file, we can have a short python program that
# generates points, one per line.
# The following string, when interpreted as a datafile,
# generates 1001 points from 400 to 40e3, with uniform ratio between them
fit_points = "< python -c 'for i in range(1001): print 400.*(10.**((i*0.02))'"
## generates 1001 points from 400 to 4e6, with uniform ratio between them
# fit_points = "< python -c 'for i in range(1001): print 400.*(10.**((i*0.04))'"

# Pick a target value for the constant impedance at high frequency
# (say the resistance at 1kHz, which is in the flat region)
Rtarget=abs(Zloud(1e3))

# Alternately fit Ccomp and Rcomp to try to keep the impedance near Rtarget
# (the scales are too different to fit both at once).
fit abs(Zcomp(x,Rcomp,Ccomp)) fit_points using 1:(Rtarget) via Ccomp
fit abs(Zcomp(x,Rcomp,Ccomp)) fit_points using 1:(Rtarget) via Rcomp
fit abs(Zcomp(x,Rcomp,Ccomp)) fit_points using 1:(Rtarget) via Ccomp
fit abs(Zcomp(x,Rcomp,Ccomp)) fit_points using 1:(Rtarget) via Rcomp

Rround=8.2
Cround=3.3e-6

plot abs(Zloud(x)) title "uncompensated loudspeaker",\
      abs(Zcomp(x,Rcomp,Ccomp)) title sprintf("loudspeaker || (%.3g ohm + %.3g F)",Rcomp,Ccomp), \
      abs(Zcomp(x,Rround,Cround)) title sprintf("loudspeaker || (%.3g ohm + %.3g F)",Rround,Cround)

```

Figure 5: Gnuplot script to choose compensation resistor and capacitor to make the loudspeaker have a roughly constant impedance at frequencies above 1kHz. The last plot was added manually to plot the compensation after rounding to the nearest standard component values. This script will be made available at <http://users.soe.ucsc.edu/~karplus/bme194/w13/lab-handouts/loudspeaker-compensation-power-law.gnuplot>

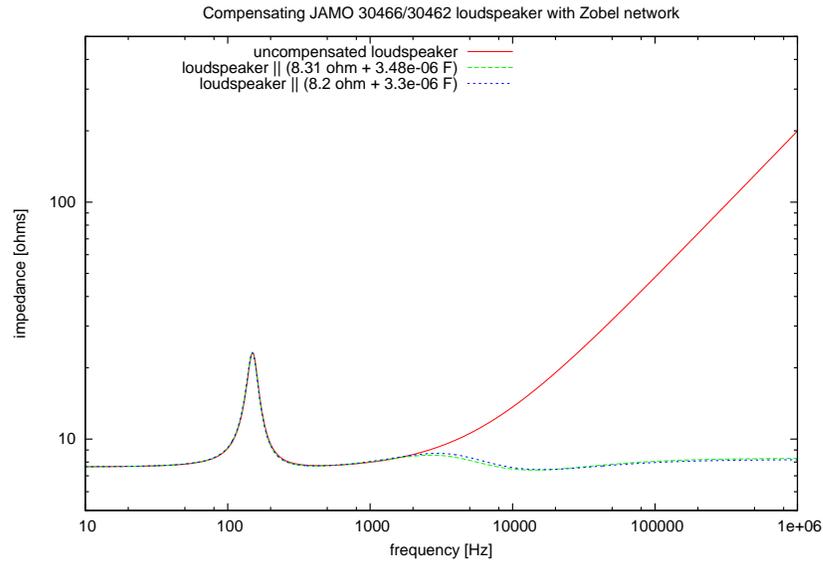


Figure 6: The Zobel network (R+C in parallel with the loudspeaker) looks like it would do a good job of keeping the impedance nearly constant for high frequencies, assuming that our model of the loudspeaker is reasonable.

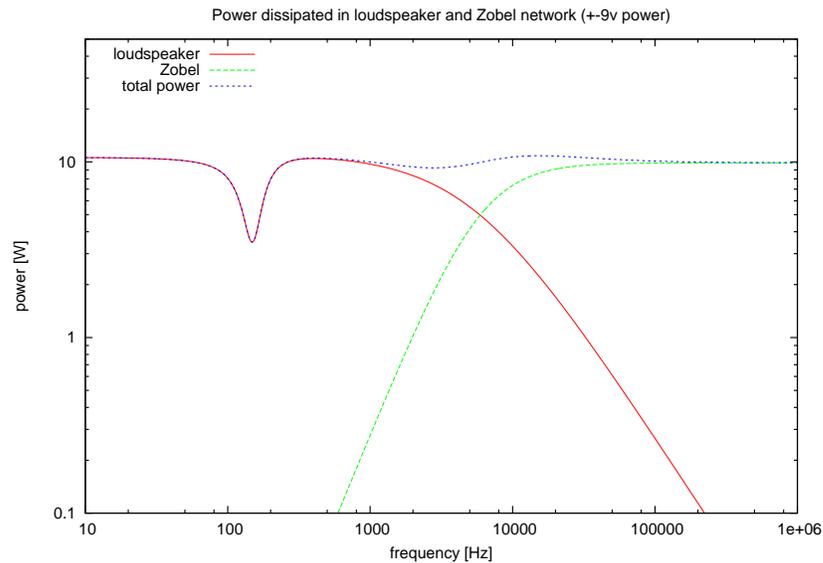


Figure 7: The Zobel network dissipates a lot of power at high frequencies, so the amplifier sees a constant load, even though the loudspeaker is not getting much power.

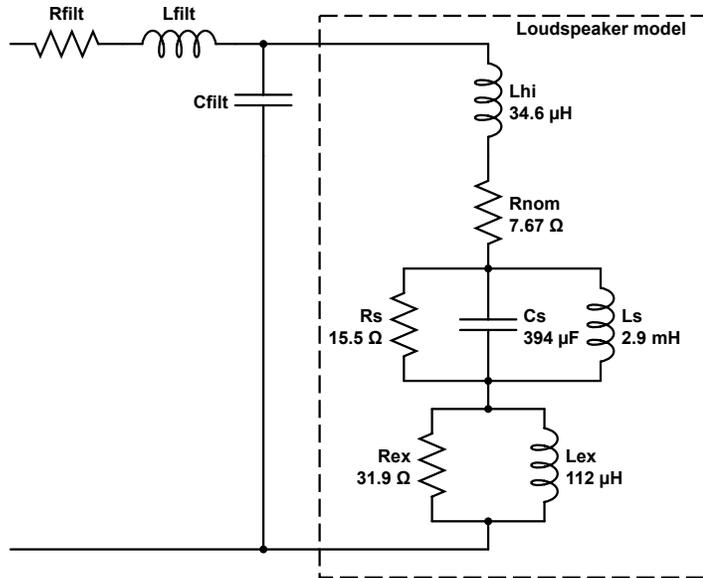


Figure 8: An LC low-pass filter combined with the loudspeaker model, without a Zobel network for compensating the speaker inductance. The  $R_{filt}$  resistance is not a resistor that you add, but a model for the resistance of the wire in the inductor.

## 2.7.2 Output filter without Zobel network

Can you design the LC low-pass filter directly, without a Zobel network to make the speaker look like a simple resistive load (like in Figure 8)?

Sure! you can use essentially the same trick we used to design the Zobel network, asking gnuplot to fit parameter values for  $L_{filt}$  and  $C_{filt}$  to make the gain of the voltage divider be nearly constant up to the highest frequency you are interested in, then drop off rapidly. A gnuplot script that does that is shown in Figure 9. There are lots of parameters in the script that can be changed (like what the desired gain is as a function of frequency and whether the fitting should be done with additive

Initially, I was going to have students wind their own inductors around cardboard tubes, picking the number of turns of wire using an inductance calculator like the ones at <http://www.pronine.ca/multind.htm> and

<http://www.crystalradio.net/professorcoyle/professorcoylecyl.shtml>

But after figuring out how much wire would be needed and making an attempt to wind one myself, I decided it would be better and cheaper if I just bought some inductors for the class. I ended up buying a  $220\mu\text{H}$  inductor with a  $0.252\Omega$  resistance and a maximum current of 1.9A (the AIUR-06-221K made by Abracon Corporation). These are high-power ferrite-core inductors, but they have one problem: their leads are too fat for bread boards, so you'll have to attach them to your circuit with clip leads.

To make design easier for you, I've also written a gnuplot script (Figure 10) that plots the real power into the loudspeaker if you use this inductor with various standard capacitor sizes. It even allows you to set the value for a small resistor in series with the loudspeaker for viewing current waveforms on the oscilloscope. (Such a resistor does change the LC design somewhat.)

```

set title "Gain for LC filter driving JAMO 30466/30462 loudspeaker"
set logscale xy
set samples 1000
set key bottom left
set xlabel "frequency [Hz]"
set xrange[1:1e6]
set ylabel "gain"
set yrange [0.0005: 5]

j=sqrt(-1)
conjugate(a) = real(a) - j*imag(a)
phase(v) = imag(log(v))
real_power(V,Z)= real(V*conjugate(V/Z)) # V is RMS voltage phasor

Zc(f,C) = 1/(j*2*pi*f*C) # f= frequency [Hz], C=cap [farads]
Zl(f,L) = j*2*pi*f*L
Zpar(z1,z2) = z1*z2/(z1+z2)
divider(Zup,Zdown) = Zdown/(Zup+Zdown)

#impedance of loudspeaker
Zloud(f) = Rnom+Zl(f,Lhi*f**Lhipow)+Zpar(Rs,Zpar(Zl(f,Ls),Zc(f,Cs)))
Rnom=7.64;
Lhi= 5.88e-3; Lhipow = -0.378
Ls=2.92e-3; Cs=386e-6; Rs=15.5

# These are very rough approximations to the DC resistance of an
# air-wound coil, based on estimated wire length (45mm diam core, 200microhenries)
Rfilt(Lfilt) = 1800*Lfilt # approx DC resistance with 22 gauge wire
# Rfilt(Lfilt) = 2800*Lfilt # approx DC resistance with 24 gauge wire
# Rfilt(Lfilt) = 4400*Lfilt # approx DC resistance with 26 gauge wire
# Rfilt(Lfilt) = 13500*Lfilt # approx DC resistance with 30 gauge wire

# LC divider with loudspeaker load
gain(f,Lfilt,Cfilt) = divider(Zl(f,Lfilt)+Rfilt(Lfilt),Zpar(Zloud(f),Zc(f,Cfilt)))
Ztotal(f,Lfilt,Cfilt) = Zl(f,Lfilt)+Rfilt(Lfilt) + Zpar(Zloud(f),Zc(f,Cfilt))

cutoff =20000
desired(f) = f<cutoff? 0.95: 0.01 # choose step
# desired(f) = 0.9*(f<cutoff? 1: 0.2*(cutoff/f)**2) # choose step*f^-2

# error_model(y)=log(y) # fit at low gain as well as high
error_model(y)= y # fit mainly in the passband (high gain)

fit_points = "< python -c 'for i in range(100): print 400.*(10.**i*0.04)'"

Cfiltp1 = 0.1e-6; Lfiltp1 = 100e-6
fit error_model(abs(gain(x,Lfiltp1,Cfiltp1))) fit_points \
using 1:(error_model(desired($1))) via Lfiltp1

Cfilt1 = 1e-6; Lfilt1 = 100e-6
fit error_model(abs(gain(x,Lfilt1,Cfilt1))) fit_points \
using 1:(error_model(desired($1))) via Lfilt1

Lfilt2=100e-6; Cfilt2=2.0e-6
fit error_model(abs(gain(x,Lfilt2,Cfilt2))) fit_points \
using 1:(error_model(desired($1))) via Lfilt2

Cfilt2p2 = 2.2e-6; Lfilt2p2 = 100e-6
fit error_model(abs(gain(x,Lfilt2p2,Cfilt2p2))) fit_points \
using 1:(error_model(desired($1))) via Lfilt2p2

Lfilt3=100e-6; Cfilt3=3.3e-6
fit error_model(abs(gain(x,Lfilt3,Cfilt3))) fit_points \
using 1:(error_model(desired($1))) via Lfilt3

Cfilt4=4.7e-6; Lfilt4=100e-6
fit error_model(abs(gain(x,Lfilt4,Cfilt4))) fit_points \
using 1:(error_model(desired($1))) via Lfilt4

Cfilt10=10e-6; Lfilt10=100e-6
fit error_model(abs(gain(x,Lfilt10,Cfilt10))) fit_points \
using 1:(error_model(desired($1))) via Lfilt10

plot desired(x) title "desired", \
abs(gain(x,Lfiltp1,Cfiltp1)) title sprintf("L=%.3g uH C=%.3g uF", Lfiltp1*1e6, Cfiltp1*1e6), \
abs(gain(x,Lfilt1,Cfilt1)) title sprintf("L=%.3g uH C=%.3g uF", Lfilt1*1e6, Cfilt1*1e6), \
abs(gain(x,Lfilt2,Cfilt2)) title sprintf("L=%.3g uH C=%.3g uF", Lfilt2*1e6, Cfilt2*1e6), \
abs(gain(x,Lfilt2p2,Cfilt2p2)) title sprintf("L=%.3g uH C=%.3g uF", Lfilt2p2*1e6, Cfilt2p2*1e6), \
abs(gain(x,Lfilt3,Cfilt3)) title sprintf("L=%.3g uH C=%.3g uF", Lfilt3*1e6, Cfilt3*1e6), \
abs(gain(x,Lfilt4,Cfilt4)) title sprintf("L=%.3g uH C=%.3g uF", Lfilt4*1e6, Cfilt4*1e6), \
abs(gain(x,Lfilt10,Cfilt10)) title sprintf("L=%.3g uH C=%.3g uF", Lfilt10*1e6, Cfilt10*1e6)

```

Figure 9: Gnuplot script to choose inductor and capacitor values to provide low-pass filtering to the speaker. The gain at the frequency of the triangle wave should be at most about 0.01, and the gain in the passband (up to the highest frequency you want to amplify) should be fairly flat. This script will be made available at <http://users.soe.ucsc.edu/~karplus/bme194/w13/lab-handouts/loudspeaker-low-pass-design-power-law.gnuplot>

```

set logscale xy
set xlabel "frequency [Hz]"
set xrange[10:1e6]
set key bottom left Left
set samples 1000
set grid

Rseries = 0 # extra resistor for displaying current on scope
V_supply=3.3 # power supply
V_RMS = V_supply/sqrt(2) # RMS voltage for largest sine wave from supply
set title sprintf("Low-pass LC filter for loudspeaker with %.3gohm series resistor",Rseries)
set yrange [1e-4:40]
set ylabel sprintf('Real power to speaker with +- %.3gV supply [W]',V_supply)

j=sqrt(-1)
conjugate(a) = real(a) - j*imag(a)
phase(v) = imag(log(v))
real_power(V,Z)= real(V*conjugate(V/Z)) # V is RMS voltage phasor

divider(Zup,Zdown) = Zdown/(Zup+Zdown)
Zc(f,C) = 1/(j*2*pi*f*C) # f= frequency [Hz], C=cap [farads]
Zl(f,L) = j*2*pi*f*L
Zpar(z1,z2) = z1*z2/(z1+z2)

#model for impedance of loudspeaker
Zloud(f) = Rnom+Zl(f,Lhi)+Zpar(Rs,Zpar(Zl(f,Ls),Zc(f,Cs))) + Zpar(Rex,Zl(f,Lex))
Rnom=7.67
Lhi=34.6e-6
Ls=2.9e-3; Cs=394e-6; Rs=15.5
Lex=112e-6; Rex=31.9

# voltage divider with Lfilt and Cfilt
gain(f,Lfilt,Rfilt,Cfilt) = divider(Zl(f,Lfilt)+Rfilt, Zpar(Zloud(f)+Rseries,Zc(f,Cfilt)))

# total impedance, in case we want to plot that also
Ztotal(f,Lfilt,Rfilt,Cfilt) = Zl(f,Lfilt)+Rfilt + Zpar(Zloud(f)+Rseries,Zc(f,Cfilt))

# real power into loudspeaker with voltage from amplifier V at frequency f
loud_power(V,f,Lfilt,Rfilt,Cfilt) = real_power(V*gain(f,Lfilt,Rfilt,Cfilt)*divider(Rseries,Zloud(f)), Zloud(f))

Lfilt=220e-6; Rfilt=0.252 # parameters for AIUR-06-221 inductor
half_power=0.5*loud_power(V_RMS,1000,Lfilt,Rfilt,3.3e-6) # half-power point

plot half_power title sprintf("half-power=%.3gW",half_power), \
loud_power(V_RMS,x,Lfilt,Rfilt,0.1e-6) title sprintf("L=%.3guH C=%.3guF", Lfilt*1e6, 0.1) \
loud_power(V_RMS,x,Lfilt,Rfilt,0.2e-6) title sprintf("L=%.3guH C=%.3guF", Lfilt*1e6, 0.2) \
loud_power(V_RMS,x,Lfilt,Rfilt,1e-6) title sprintf("L=%.3guH C=%.3guF", Lfilt*1e6, 1) \
loud_power(V_RMS,x,Lfilt,Rfilt,2.2e-6) title sprintf("L=%.3guH C=%.3guF", Lfilt*1e6, 2.2) \
loud_power(V_RMS,x,Lfilt,Rfilt,3.3e-6) title sprintf("L=%.3guH C=%.3guF", Lfilt*1e6, 3.3) \
loud_power(V_RMS,x,Lfilt,Rfilt,4.7e-6) title sprintf("L=%.3guH C=%.3guF", Lfilt*1e6, 4.7) \
loud_power(V_RMS,x,Lfilt,Rfilt,10e-6) title sprintf("L=%.3guH C=%.3guF", Lfilt*1e6, 10)

```

Figure 10: Gnuplot script to choose capacitor values to provide low-pass filtering to the speaker. The power to the loudspeaker should be very low at the PWM frequency, fairly constant in the range of frequencies you want to amplify, and below the 10W limit for the speaker at all frequencies. You'll have to play with the script to adjust the power supply voltage and the series resistor. This script will be made available at <http://users.soe.ucsc.edu/~karplus/bme194/w13/lab-handouts/loudspeaker-low-pass-choose-C.gnuplot>

### 3 Pre-lab assignment

Select a partner at least 2 days before the lab and start working with them.

Make some design specifications for your amplifier: what is the frequency range you wish to amplify? what is the power output of the amplifier? What AC voltage or current does you input have?

What frequency do you want your PWM signal to be at? It should be at least 3 times the frequency of the highest frequency you want to amplify and above the range of human hearing. If it is too high, the FETs will be spending a greater portion of their time in the intermediate state where they are neither on nor off, and the efficiency of the amplifier will be reduced (and the FETs might get hot).

As you did for the one-op-amp audio amplifier lab, make a block diagram of the whole amplifier, from microphone to loudspeaker, giving the function of each block and any constraints (voltage, current, frequency, . . . ) on the signals between the blocks. What are the voltage (or current) levels (DC and AC) for every signal that goes between blocks? Remember to include blocks for the power supplies and for the function generator.

Your preamplifier using the op-amp chip should be powered from the 6v single supply, while the comparator, CMOS output stage, and loudspeaker should be powered from the dual 9v supply. Think about how these separate voltage sources relate to one another, and make sure you include them in your schematics.

If you need to change DC voltage levels from one block to another, remember that you can use high-pass filters to do the conversion (as we did between the microphone and the op amp in the one-op-amp amplifier).

Think about how you can control the gain of the amplifier, since it might be too loud and no one wants to listen to feedback squeal. If your pre-amp stage has too much gain, you may get clipping and distortion, even if the overall loudness is reasonable. There are at least three very different ways to control the output volume—so try to think up more than one and figure out what the trade-offs are between them

Design each block.

Put a  $0.25\Omega$  resistor in series with your loudspeaker (two  $0.5\Omega$  resistors in parallel), so that you can use the oscilloscope to look at the current through the speaker as well as the voltage across it.

Draw a complete schematic of the entire amplifier, showing every component value and every pin number. You will be building and debugging from this schematic, so it must be as complete and accurate as you can make it.

### 4 Parts, tools, and equipment needed

**Parts for this lab from kit:**

- electret microphone
- MCP6004 quad op amp
- LM2903 dual comparator
- NTD5867NL-1G nFET 18A
- NTD2955-1G pFET 12A
- 10W  $8\Omega$  loudspeaker
- breadboard
- various resistors and capacitors

**Equipment in lab:**

- function generator (for triangle wave)
- oscilloscope for observing voltage and current waveforms
- 6v single and  $\pm 9V$  dual power supplies

## 5 Procedures

Build and test your circuit a block at a time. It is very hard to debug a large circuit when the problem could be anywhere.

It might be a good idea to build all the 6V circuitry (the preamp) on one breadboard and all the  $\pm 9V$  circuitry (the power amp) on a different breadboard. There should then be only 3 wires between the boards: the two inputs to the comparator and a reference voltage to make sure that those two inputs remain within the power rails of the comparator.

It is a good idea to build the preamp starting forward from the microphone, testing each block as you add it.

It is a good idea to build the power amp backwards, starting from the loudspeaker, again testing each block as you add it. It is also a good idea to start out the power amp section with a lower voltage and current limit on the power supply than you intend to finally use it, to avoid overheating components before the wiring has been debugged.

## 6 Demo and writeup

Demonstrate your amplifier working, producing undistorted amplified speech. Show the output waveforms (current and voltage) on the oscilloscope. It is worthwhile to show the waveforms with current and voltage as the two channels, so the phase relationship between the two can be seen. It is then useful to observe it at two different scales: one where the PWM waveform is clearly visible, maybe  $2\mu s$ /division, and one where the audio output is clearly visible, maybe 1ms/division.

Document both the system-level design and the details of each block in your report.

## References

- [1] Kevin Karplus. Better model for loudspeaker. 26 Feb 2013. <http://gasstationwithoutpumps.wordpress.com/2013/02/26/better-model-for-loudspeaker/>
- [2] Kevin Karplus. Seventeenth day of circuits class: inductors and gnuplot tutorial. 15 Feb 2013. <http://gasstationwithoutpumps.wordpress.com/2013/02/15/seventeenth-day-of-circuits-class-inductors-and-gnuplot-tutorial/>
- [3] W. Marshall Leach. *Introduction to Electroacoustics and Audio Amplifier Design*, chapter The Class-D Amplifier. Kendall/Hunt, second edition—revised printing edition, 2001. <http://users.ece.gatech.edu/mleach/ece4435/f01/ClassD2.pdf>
- [4] John L. Murphy. Neutralizing L(e) with a Zobel. Tech topics, topic number 8, Updated: Monday, 26 September 2011. [http://www.trueaudio.com/st\\_zobel.htm](http://www.trueaudio.com/st_zobel.htm)
- [5] Wikipedia. Class-d amplifier. read 17 Feb 2013. [http://en.wikipedia.org/wiki/Class\\_d\\_amplifier](http://en.wikipedia.org/wiki/Class_d_amplifier)
- [6] Wikipedia. Pulse-width modulation. read 17 Feb 2013. [http://en.wikipedia.org/wiki/Pulse-width\\_modulation](http://en.wikipedia.org/wiki/Pulse-width_modulation)