**Design and simulation of passive filters**

Filters are circuits that allow the desired frequency range of an input signal to pass through them while attenuating the rest. Filters can be realized with passive components such as resistors, capacitors and inductors (passive filters) or active components such as transistors and op-amps (active filters). Since active components are inherently amplifying devices, they provide gain. Thus, active filters strengthen the input signal in their pass band, as they filter out the unwanted signal. However, passive filters have no gain, so the output signal is smaller than the input signal.

Depending on the frequency range that they allow to pass through them, there are different kinds of filters:

**Low Pass Filter (LPF):**

A first order LPF can be constructed with a single resistor and a capacitor as shown below:

The gain at pass band is unity, while the signals above the cut-off frequency, are attenuated. Since there is a single pole, the attenuation slope is -20dB/dec. Sometimes we need a **steeper** filter characteristic. In such cases, we need to use a higher-order filter. A higher order filter can be realized by **cascading,** i.e. by connecting two first-order filters back to back.

Remember that, double poles give rise to a slope of -40dB/dec!

**1.**  Connect two first-order LPFs together to form a second-order LPF as shown below and show that the transfer function is given as in (1).



**2.** The transfer function can also be written as follows:

Since the second stage, loads the first stage, we can’t simply say that:

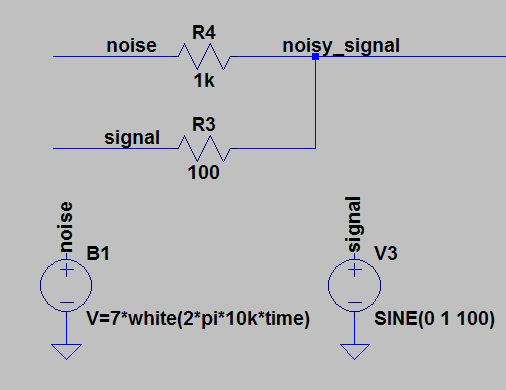
However, if the impedance seen by the first stage is large enough, then the transfer function can be approximated as in (3). Design a second-order RC LPF to have double poles at . Note that you want and (then what would be ) so that the second stage won’t load the first stage as much, and the transfer function in (3) can be used to approximate Implement the circuit in SPICE using standard resistor and capacitor values. First chose and (which means ) and perform AC analysis, plot the output. Using the graph, measure the magnitude at . Then chose and (so that ) and do the same. Comment on the different magnitude values at . Are the magnitude and phase responses as expected?

**3.** LPFs are used in audio applications to generate the “bass” signal which is the low frequency audio. The output of the LPF in those applications drive the sub-woofer – a loudspeaker for low frequency audio. For such an application the cut-off frequency is generally around 80[Hz]. Another application of an LPF is in generation of an analog signal from a Pulse Width Modulated (PWM) signal. Some transducers produce a PWM output. Using an LPF, this PWM signal can be converted to an analog signal. The value of the analog signal changes as the duty cycle of the PWM signal changes.

Now, use the LPF filter that you designed above, to convert a PWM signal to an analog signal. To do that, set the input signal to the LPF to be a pulse signal with a frequency of 8[kHz] with VON=5[V] and Tdelay=1[ns], i.e. PULSE(0 5 1ns 0 0 0.06ms 0.125ms). Apply transient analysis up to 10[ms] and plot the output voltage. Comment on the graph. Now change the duty cycle which is the ratio of the signal high time (0.06ms) to the period (0.125ms), by changing the high time to 0.1ms. Apply transient analysis. What do you observe at the output?

**4.** Through filtering, we can also get rid of noise. If you have unwanted noise (high frequency) on a supply voltage feeding into a circuit, you can connect a capacitor in parallel from the supply voltage to the ground to reduce the noise. That’s one of the most common applications of an LPF.

Now use the second-order LPF that you designed to filter out high frequency noise added to a 100[Hz] signal. We will generate a noisy signal at the input of the filter as shown below. Note that, we can use a behavioral voltage source (bv) in SPICE to generate white noise. Now, apply transient analysis and plot the noisy input signal and the filtered output signal. Comment on the result.



**High Pass Filter (HPF):**

In audio applications, HPFs are used to filter out the low frequency bass signals and direct high frequency audio signals to a tweeter. The cut-off frequency for a tweeter is somewhere around 2[kHz] to 3[kHz]. A first order HPF can be constructed with a single resistor and a capacitor as shown below:



Again, the gain at passband is unity. The frequencies below the cut-off frequency, are attenuated. If you want a steeper roll-off at low frequencies, you need to add a second pole to get a second-order HPF.



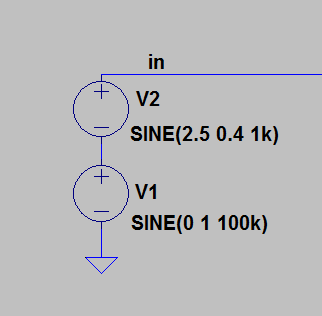
**5.** For the second-order HPF given above, show that the transfer function is given as in (4).

**6.** Design the second-order HPF for . As in the case of designing an LPF, if you chose the second stage impedances high (10 first stage impedances), the second stage won’t load the first stage and the cut-off frequency can be estimated as double poles at:

Apply AC analysis and verify the frequency response of the filter.

**7.**  One very common use of an HPF is in blocking DC component in a circuit. When we **cascade** two amplifier stages, if you do not want the DC output voltage of one stage to affect the following stage, we put a series capacitor in between the two stages. Called a **coupling capacitor**, this capacitor lets the AC signal to pass through, while blocking the DC signal.

Now to the input of your HPF apply a signal that is composed of two sinusoids with different frequencies added together, as shown below. Apply transient analysis up to 5[ms] and plot both the input and output of the filter. Explain what happens.



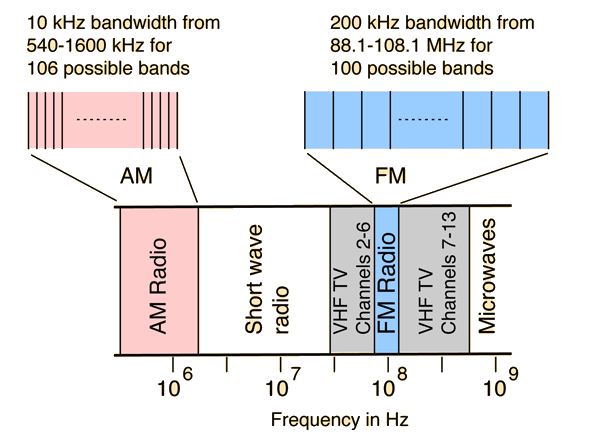
**Band Pass Filter (BPF):**

A BPF can simply be realized by connecting an LPF together with an HPF.

**8.** Connect a first-order HPF to a first-order LPF to get a BPF. Design the filter to have a low-frequency cut-off frequency of and a high frequency cut-off frequency of This means a bandwidth of approximately Note that, LPF stage will determine the of the BPF, while HPF stage determines the of the BPF. As before, if you chose the resistor and impedance of the capacitor of the LPF (second stage in this case) higher than the HPF’s (first stage), you can estimate and as follows:

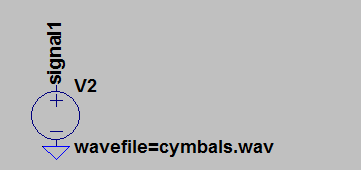
Apply AC analysis and obtain the magnitude and the phase plot of the BPF. Are the results as expected?

**9.** The figure given below shows the frequency spectrum for AM and FM waves. The signal is modulated by the carrier frequency and then transmitted through electromagnetic waves. When the signal is received at the receiver end, it is accompanied by other frequency components at the adjacent bands. In order to get the desired frequency band, we need to first band pass filter the received signal at the antenna. Once filtered, we can **demodulate** the signal and then use a LPF to recover the original signal.

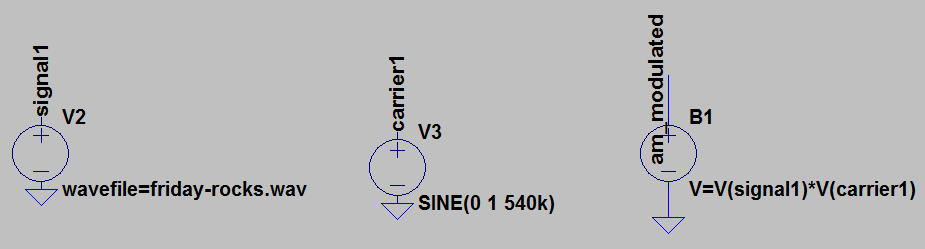


For the last part of this assignment, we will work with audio files. Perform the following steps:

* First save the provided *.wav* files into the same LTSpice folder as your circuits. Then use the statement *wavefile=audio\_file.wav* to read in the *.wav* files to be used into your circuit. Generate three different signals, associated with three different audio files. Apply transient analysis. I tried to find short audio files, however you will still need to run transient analysis up to 2[s]. Plot these signals as a function of time.



* Now you will need to create an Amplitude Modulated (AM) signal. To do that, we will use the behavioral voltage source again. Simply, multiply the audio signal with the carrier signal which is pure sine wave. Multiplication in time domain, means convolution in frequency domain. Select the carrier frequency as 540[kHz]. Apply transient analysis and observe the AM signal.



* Now, you will need to create two such AM signals, representing two different radio channels. The goal is to recover one of the audio signals at the receiver end. This will be the radio channel that you want to listen to. The first step to achieve this goal, is to use a BPF to select the channel of interest. You will need a **narrowband** BPF. Since the BPF designed above is **wideband**, i.e. have a large bandwidth, we will use another one which is one of the LTSpice demo circuits provided with this homework. Apply AC analysis to the narrowband BPF provided and observe its frequency response.
* Now, select two of the audio files provided. Generate two AM signals by modulating the two selected audio files by two different carrier frequencies. The choice for the carrier frequencies is yours. AM carrier frequencies range from 540[kHz] to 1600[kHz]. (Make sure they are at least 200[kHz] apart.) Then add the two AM signals (connect in series) and apply to the input of the filter. Note that the filter’s center frequency is tunable. By setting the parameter you can change the center frequency of the BPF. must be set to the AM carrier frequency that you desire to detect.
* If done successfully, at the output of the filter, you should only have the desired channel, i.e. the desired AM signal while attenuating the other. But the original audio signal is not recovered yet. You need to **demodulate** the AM signal by multiplying it with the same carrier frequency. To do that, you need to multiply the output voltage of the BPF with the voltage of the carrier signal by means of a behavioral voltage source (as you did to generate the AM signal).
* Multiplying a signal with a sinusoid shifts that signal down to DC and also up to . In order to choose the signal at DC, apply the signal to a first order RC LPF with. At the end, you should get one of the original audio signals that you applied at the input.
* Last step is to listen to the recovered audio signal. To do that use SPICE directive to write the statement:

.wave "C:\Users\Gulin\Desktop\recovered\_audio.wav" 16 44.1k V(recovered)

When you click on the recovered\_audio, you should hear the original sound.

Here is a block diagram of what we did:



Once you construct all of the building blocks shown above, you will be applying transient analysis. Choose the duration depending on the length of the original audio signal that you want to recover. For instance, for “Friday-rocks.wav” 1.6[s] should be enough. Be aware that this final simulation takes a long time – may be more than couple of hours.

Provide a plot of the original input signal and the recovered signal. Prove to me and yourself that the shape of the output waveform is very similar to the shape of the desired audio waveform. Comment on all of the plots that you provide.

With couple of sentences, summarize, what you’ve learnt from this project!