Analog and Digital Filters

Anthony Garvert
November 13, 2015

Abstract

In circuit analysis and performance, a signal transmits some form of information, such as a voltage or current. However, over a range of frequencies, not all of this signal is desirable. In audio circuits, for instance, factors such as ambient noise can affect what one would want to actually measure, record, and output. Thus, these extraneous portions of the signal should be cut out. This is done using a type of circuit known as a *filter*, which will pass and potentially amplify a certain desirable frequency signal, while simply attenuating the signal in other frequency regions.

Keywords

To properly learn about filter design, several key concepts must be covered. The first and most important concept is known as the *gain* of the circuit. This is simply the ratio of the output voltage or current to the input voltage or current. This gain depends on frequency, due to the frequency-variant nature of capacitors and inductors. The next important concept to discuss *bode plots*. A bode plot will plot the magnitude and phase of the circuit gain, H(s), against the frequency of the circuit, on a logarithmic frequency scale.

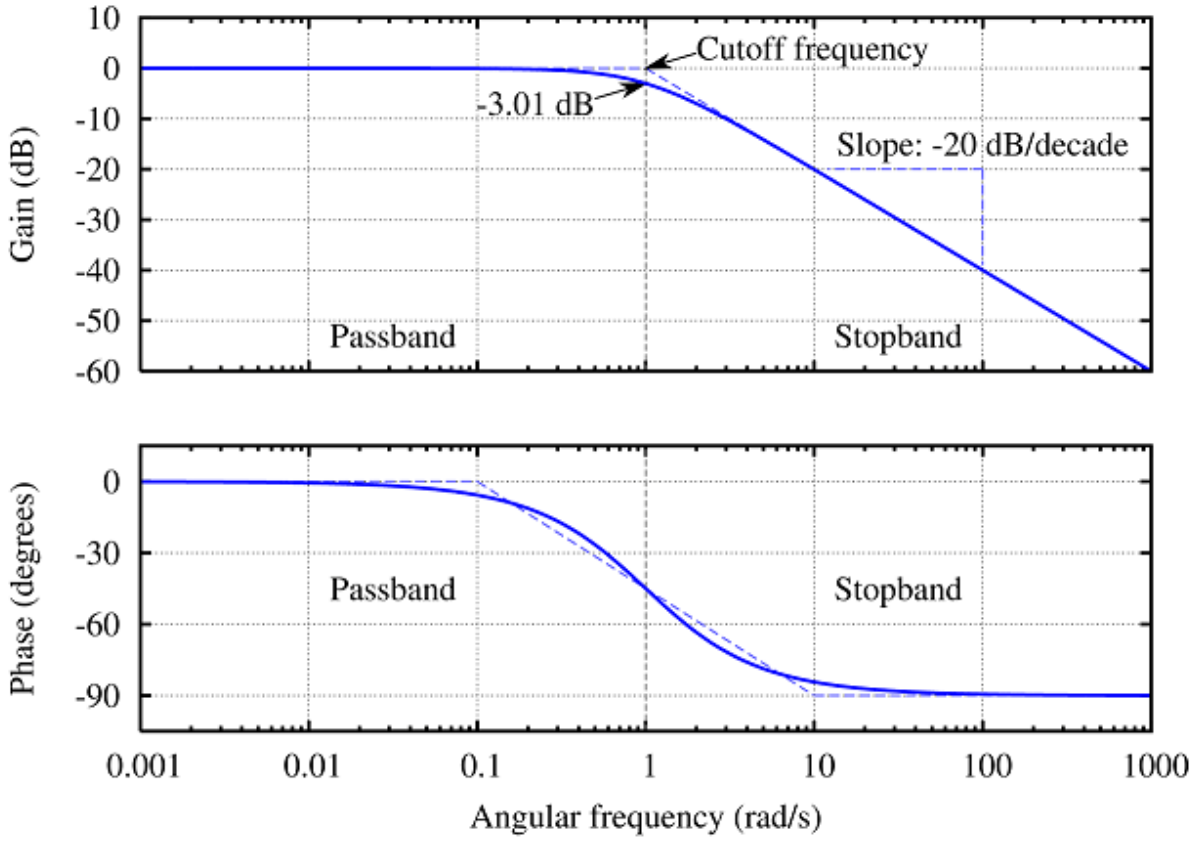


Figure 1. An example of the magnitude (above) and phase (below) of a bode plot.

Next, it is crucial to discuss the various types of filters that can be implemented. The first is a *low-pass* filter, which will allow lower-frequency signals to pass, and attenuate the high-frequency signals. The *high-pass* filter will perform the opposite function, instead allowing the higher-frequency signals to pass and attenuating the lower frequencies. Finally, the *band-pass* filter will pass a specific band of frequencies, while attenuating all signals lower and higher than that frequency band. In each of these types of circuits, there exists a specific frequency known as the *cutoff frequency*, which is the point at which the circuit gain drops 3dB from the pass band gain. This cutoff frequency represents the point where the filter transitions from passing a frequency to attenuating it. The last crucial concept to cover is the difference between analog and digital filters. *Analog filters* are created from circuit elements including resistors, inductors, capacitors, and operational amplifiers. *Digital filters* are created by converting a continuous signal into a discrete signal, and then altering it using simple mathematics (addition, subtraction, multiplication, division) and the memory storage capabilities of a microcontroller. This signal will be converted back into a continuous signal and output.

Introduction

The signal that will be filtered in this instance is an audio sample of a diamond being polished, which will be picked up by a directional microphone. The process of polishing a diamond will tend to emit sounds at certain frequencies indicating the smoothness of the diamond surface. However, there are several kinds of noise that might interfere with the measurement of the noise strictly emitted between the diamond and polishing wheel. Such noises include the ambient noise of the environment, as well as the noise created by the machine without a diamond touching its surface. However, it is still important to monitor the noise created by the machine for balance purposes. Thus, filters will be used to separate out the different kind of noise and attenuate the unwanted parts of the signal.

Objective

The main objective is to create an effective band-pass filter to focus on the noise emitted by the diamond and machine. An analog high-pass filter will be used to filter out the ambient noise (any noise less than 5 kHz) created by the environment around the diamond polishing machine. The higher end of the frequency measurement will be determined by the sensitivity of the microphone, thus creating the entire band of 5-20 kHz. After this ambient noise is filtered out, the signal will be digitally filtered to differentiate the remaining types of noise. This digital filter will separate the signal into machine noise (5-12 kHz) and the diamond noise (12-20 kHz). The following steps will detail the process of designing the corresponding filters.

Process

The first aspect to the filter design incorporates an analog filter to clip off the ambient sounds lower than 5 kHz. This analog filter can be chosen using a design from TI (Figure 3). This circuit is a Sallen-Key design with a Butterworth-type response. This design allows one to easily choose the resistor and capacitor values to achieve the desired cutoff frequency at 5 kHz. This is the "frequency" shown on the x-axis in Figure 2.

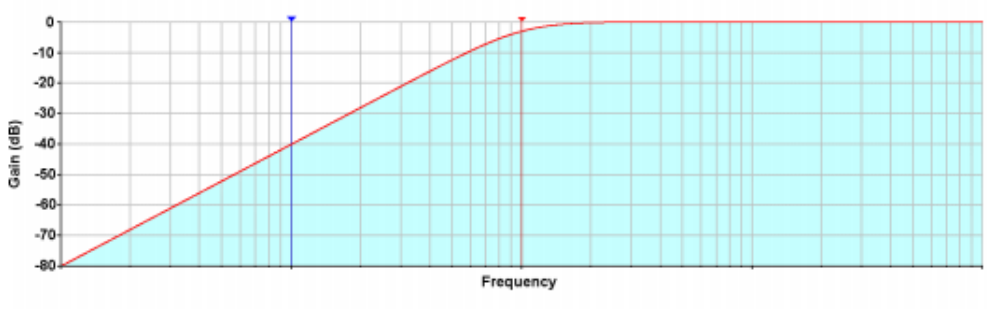


Figure 2. The bode plot of the low-pass filter, eliminating all the ambient noise and machine noise below 5 kHz.

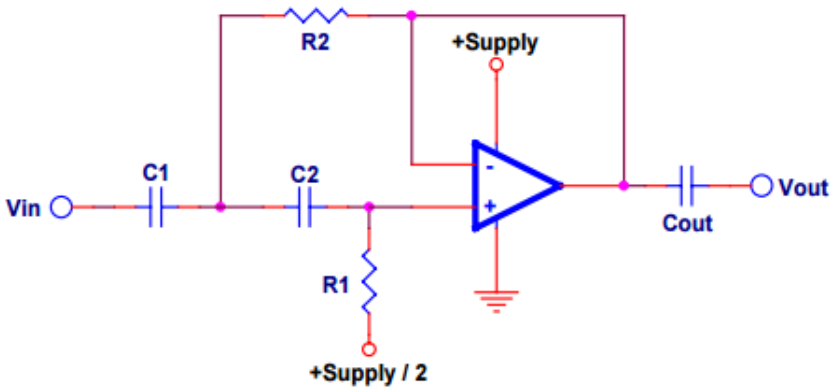


Figure 3. The circuit schematic of the high-pass analog filter, with values to be determined.

The first step is to pick values for C1 and C2 which are equal to each other. Next, the value of R1 can be chosen using the cutoff frequency and value of C1, with the equation below:

$$R1: \frac{1}{\sqrt{2} * \pi * C1 * Frequency}$$

Next, the value of R2 can be chosen using the value of C1 and the cutoff frequency:

$$R2: \frac{1}{2\sqrt{2} * \pi * C1 * Frequency}$$

Finally, Cout will be chosen as a value 100 to 1000 times C1. This capacitor will help stabilize the signal and avoid noise and rippling, especially in the pass band. It is important to consider that the values of R1, R2, C1, and C2 should be standard values. They will contain tolerances that will vary the values slightly, which is why the signal will be filtered again using a digital filter. When this signal is filtered into two distinct regions, 5-12 kHz and 12-20 kHz, it will be done so digitally. The digital filter is implemented by using a third order lattice wave filter. This filter alters the values of the discrete signal it receives from the analog-to-digital converter (Figure 4).



Figure 4. The flowchart describing the process of filtering a signal digitally.

The microprocessor only uses simple mathematics, as well as values for an input using a program written in the SLAA331 datasheet provided by TI. This datasheet references a program which will take in the parameters for a particular type of filter, and then output the values of three variables alpha in order to implement the desired filter. The prompts for this program are shown below, as well as the input for this particular example:

Type: LP
Design: Butter
Structure: Normal
Pass Band Edge Frequency: 12,000
Stop Band Edge Frequency: 13,000
Sampling Frequency: 32,000
Pass Band Ripple: 0.5
Stop Band Attenuation: 18
Design Margin: 0.5
Number of Bits for Binary: 16

Type: HP
Design: Butter
Structure: Normal
Pass Band Edge Frequency: 12,000
Stop Band Edge Frequency: 11,000
Sampling Frequency: 32,000
Pass Band Ripple: 0.5
Stop Band Attenuation: 18
Design Margin: 0.5
Number of Bits for Binary: 16

In this case, this will filter the signal digitally after the analog filter has attenuated everything lower than 5 kHz. The low pass filter will pass the frequencies ranging from 5-12 kHz, which will be generated by the machine and spinning wheel. The high-pass filter will attenuate the machine noises from 5-12 kHz, and will pass instead the 12-20 kHz frequencies. However, the upper end of this band is determined by the sensitivity of the microphone, as many microphones today have a sensitivity of 16 kHz or less. The program above will output alpha values based on what it calculates the order of the circuit to be. The order is determined by how many inputs are used in the simple mathematics. For a third order circuit, it will take two previous inputs that are stored in the memory, and use the current input as the third.

These alpha values are plugged into the code shown below, and the high-pass and low-pass filters are created after altering the values. The variables can be stored in a microcontroller such as an Arduino or MSP430.

```
alpha0 = 0.121715545654297;
alpha1 = 0.120417892932892;
alpha2 = 0.022688925266266;

inp1 = voltageRaw;
inp2 = delay0;
p1 = inp1 - inp2;
outp2 = (alpha0 * p1) + inp2;
```

```
delay0 = outp2;
outp1 = outp2 - p1;
topout = outp1;

inp1 = delay1;
inp2 = delay2;
p1 = inp1 - inp2;
outp2 = (alpha2 * p1) + inp2;
delay2 = outp2;
outp1 = outp2 - p1;

inp1 = voltageRaw;
inp2 = outp1;
p1 = inp2 - inp1;
outp2 = (alpha1 * p1) - inp2;
delay1 = outp2;
outp1 = outp2 - p1;
botout = outp1;

lpfout = ((topout + botout)/2);
hpfout = ((topout - botout)/2);
```

Results/Conclusion

The final product of this design will be a band-pass filter separated into two portions for each of the two noises that will be analyzed. The frequencies will be sampled at a certain rate and output to be analyzed. It is important to consider the noise created by the filters that might distort the signal. Equally significant is the sampling rate, which must be at least twice the bandwidth of the signal. The final factor that must be considered is the rate of attenuation, which might alter the cutoff frequency. The Butterworth filter effectively combats this issue, but there are filters that exist which have an even steeper attenuation rate but will cause more ripple in the pass band of the signal.

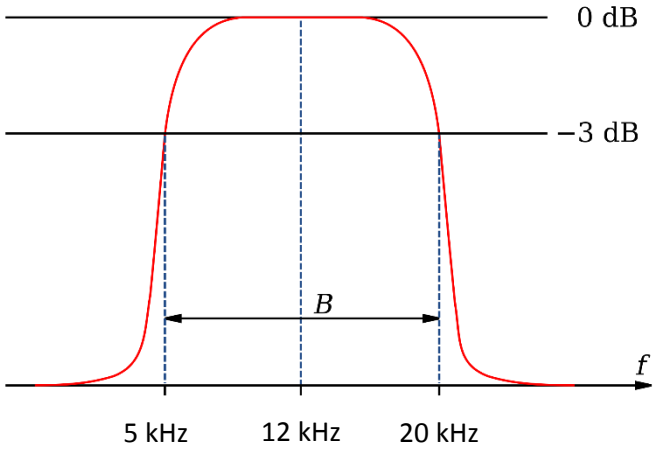


Figure 6. The final design of the bandpass filter, with the sound less than 5 kHz being ambient noise and the sound greater than 20 kHz being inaudible.

References

1. Analog Devices, Inc, “Analog Filters,” **Linear Circuit Design Handbook**. Ch. 8. Newnes, 2008.

2. Bierl, Lutz, “MSP430 Mixed Signal Microcontroller,” **SLAS735J**, Texas Instruments, 2013.

3. Venkat, Kripasagar, “Wave Digital Filtering Using the MSP430,” **SLAA331**, Texas Instruments, 2006.

4. Carter, Bruce, “Filter Design in Thirty Seconds,” **SLOA093**, Texas Instruments, 2001.