Lecture 29: Notch Filters

Mark Hasegawa-Johnson

ECE 401: Signal Processing

Outline

Review: Poles and Zeros

A first-order autoregressive filter,

$$y[n] = x[n] + bx[n-1] + ay[n-1],$$

has the impulse response and system function

$$h[n] = a^n u[n] + ba^{n-1} u[n-1] \leftrightarrow H(z) = \frac{1 + bz^{-1}}{1 - az^{-1}},$$

where a is called the **pole** of the filter, and $-b$ is called its **zero**.

Causality and Stability

- A filter is **causal** if and only if the output, $y[n]$, depends only an **current and past** values of the input, $x[n], x[n-1], x[n-2], \ldots$.
- A filter is **stable** if and only if **every** finite-valued input generates a finite-valued output. A causal first-order IIR filter is stable if and only if $|a| < 1$.

Review: Poles and Zeros

Suppose $H(z) = \frac{1+bz^{-1}}{1-az^{-1}}$, and $|a| < 1$. Now let's evaluate $|H(\omega)|$, by evaluating $|H(z)|$ at $z = e^{j\omega}$:

$$|H(\omega)| = \frac{|e^{j\omega} + b|}{|e^{j\omega} - a|}$$

What it means $|H(\omega)|$ is the ratio of two vector lengths:

- When the vector length $|e^{j\omega} + b|$ is small, then $|H(\omega)|$ is small.
- When $|e^{j\omega} - a|$ is small, then $|H(\omega)|$ is LARGE.

Review: Complex Numbers

Suppose that

$$p_1 = x_1 + jy_1 = |p_1|e^{j\theta_1}$$
$$p_2 = p_1^* = x_1 - jy_1 = |p_1|e^{-j\theta_1}$$

Then

- $p_1 + p_2$ is real:

$$p_1 + p_2 = x_1 + jy_1 + x_1 - jy_1 = 2x_1$$

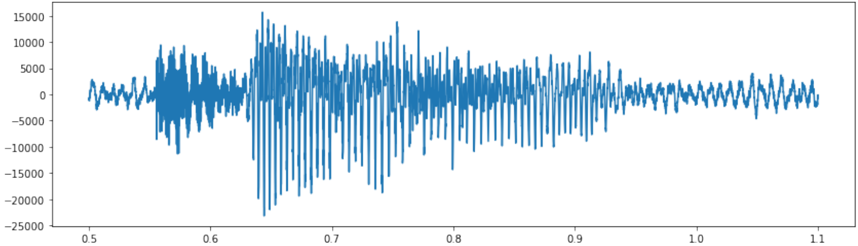
- $p_1 p_2$ is also real:

$$p_1 p_2 = |p_1|e^{j\theta_1}|p_1|e^{-j\theta_1} = |p_1|^2$$

Outline

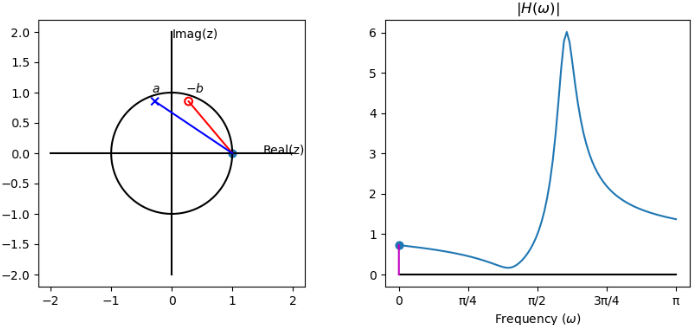
The problem of electrical noise

When your microphone cable is too close to an electrical cord, you often get noise at the harmonics of 60Hz (especially at 120Hz as shown here; sometimes also at 180Hz and 240Hz).



Can we use zeros?

As you know, zeros in $H(z)$ cause dips in $H(\omega)$. Can we use that, somehow, to cancel out noise at a particular frequency?



Can we use zeros?

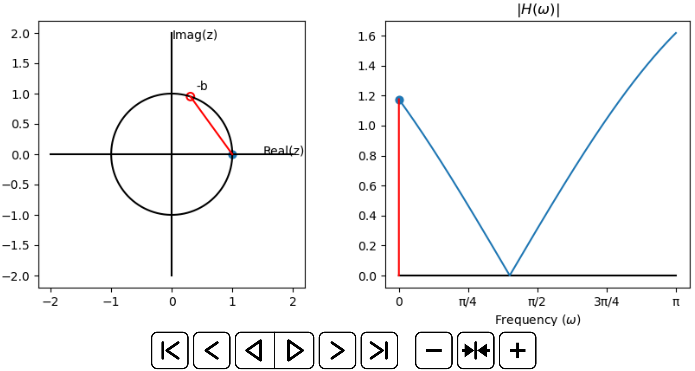
In particular,

$$H(z) = \frac{1 + bz^{-1}}{1 - az^{-1}}$$

- The pole needs to have a magnitude less than one ($|a| < 1$), otherwise the filter will be unstable, but...
- the zero doesn't have that restriction. We can set $|b| = 1$ if we want to.
- In particular, suppose we want to completely cancel all inputs at $\omega = \omega_c$. Can we just set $H(e^{j\omega_c}) = 0$?

Q: Can we just set $H(e^{j\omega_c}) = 0$?

A: YES!



Using Zeros to Cancel Line Noise

The filter shown in the previous slide is just $H(z) = 1 + bz^{-1}$, i.e.,

$$y[n] = x[n] + bx[n-1]$$

There are two problems with this filter:

1. **Complex:** b needs to be complex, therefore $y[n]$ will be complex-valued, even if $x[n]$ is real. Can we design a filter with a zero at $z = -b$, but with real-valued outputs?
2. **Distortion:** $H(z)$ cancels the line noise, but it also changes signal amplitudes at every other frequency.

Complex Conjugate Zeros

The problem of complex outputs is solved by choosing complex-conjugate zeros. Suppose we choose zeros at

$$r_1 = e^{j\omega_c}, \quad r_2 = r_1^* = e^{-j\omega_c}$$

Then the filter is

$$H(z) = (1 - r_1 z^{-1})(1 - r_2 z^{-1}) = 1 - (r_1 + r_2)z^{-1} + r_1 r_2 z^{-2},$$

but from our review of complex numbers, we know that

$$\begin{aligned} r_1 + r_2 &= 2\Re(r_1) = 2\cos(\omega_c) \\ r_1 r_2 &= |r_1|^2 = 1 \end{aligned}$$

Complex Conjugate Zeros

So the filter is

$$H(z) = (1 - r_1 z^{-1})(1 - r_2 z^{-1}) = 1 - 2\cos(\omega_c) z^{-1} + z^{-2}.$$

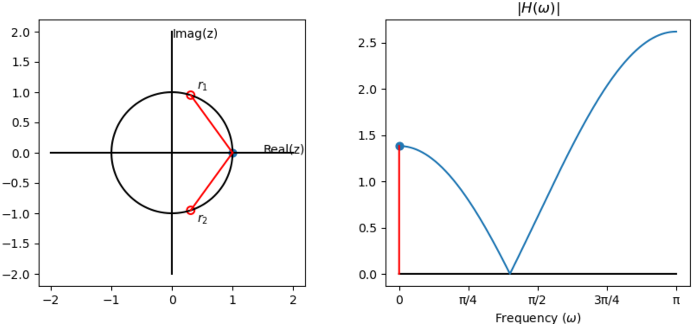
In other words,

$$y[n] = x[n] - 2\cos(\omega_c) x[n-1] + x[n-2]$$

Its impulse response is

$$h[n] = \begin{cases} 1 & n = 0 \\ -2\cos\omega_c & n = 1 \\ 1 & n = 2 \\ 0 & \text{otherwise} \end{cases}$$

Complex Conjugate Zeros



Outline

Distortion

- The two-zero filter cancels line noise, but it also distorts the signal at every other frequency.
- Specifically, it amplifies signals in proportion as their frequency is far away from ω_c. Since ω_c is probably low-frequency, $H(z)$ probably makes the signal sound brassy or tinny.
- Ideally, we'd like the following frequency response. Is this possible?

$$H(\omega) = \begin{cases} 0 & \omega = \omega_c \\ 1 & \text{most other frequencies} \end{cases}$$

Notch Filter: A Pole for Every Zero

The basic idea of a notch filter is to have a pole for every zero.

$$H(z) = \frac{1 - rz^{-1}}{1 - pz^{-1}}, \quad |H(\omega)| = \frac{|1 - re^{-j\omega}|}{|1 - pe^{-j\omega}|}$$

and then choose $r = e^{j\omega_c}$ and $p = ae^{j\omega_c}$, for some a that is very close to 1.0, but not quite 1.0. That way,

- When $\omega = \omega_c$, the numerator is exactly

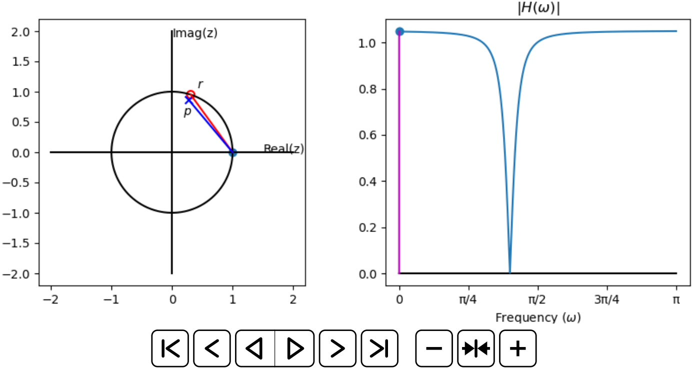
$$|1 - e^{j(\omega_c - \omega_c)}| = |1 - 1| = 0$$

- When $\omega \neq \omega_c$,

$$|e^{j\omega} - r| \approx |e^{j\omega} - p|, \quad \text{so} \quad |H(\omega)| \approx 1$$

Notch Filter: A Pole for Every Zero

The red line is $|e^{j\omega} - r|$ (distance to the zero on the unit circle). The blue line is $|e^{j\omega} - p|$ (distance to the pole inside the unit circle). They are almost the same length.



Notch Filter: Practical Considerations

Now let's consider two practical issues:

- How do you set the bandwidth of the notch?
- How do you get real-valued coefficients in the difference equation?

Bandwidth of the Notch

In signal processing, we often talk about the "3dB Bandwidth" of a zero, pole, or notch. Decibels (dB) are defined as

$$\text{Decibels} = 20\log_{10}|H(\omega)| = 10\log_{10}|H(\omega)|^2$$

The 3dB bandwidth of a notch is the bandwidth, B, at which $20\log_{10}\left|H\left(\omega_c \pm \frac{B}{2}\right)\right| = -3\text{dB}$. This is a convenient number because

$$-3 \approx 20\log_{10}\left(\frac{1}{\sqrt{2}}\right),$$

so when we talk about 3dB bandwidth, we're really talking about the bandwidth at which $|H(\omega)|$ is $\frac{1}{\sqrt{2}}$.

Bandwidth of the Notch

The 3dB bandwidth of a notch filter is the frequency $\omega = \omega_c + \frac{B}{2}$ at which

$$\frac{1}{\sqrt{2}} = \frac{|1 - rz^{-1}|}{|1 - pz^{-1}|}$$

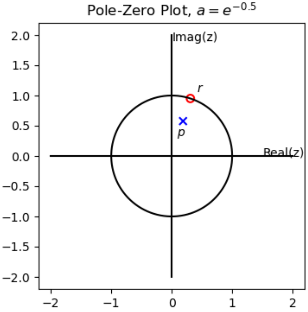
Let's plug in $z = e^{j(\omega_c + B/2)}$, $r = e^{j\omega_c}$, and $p = ae^{j\omega_c}$, we get

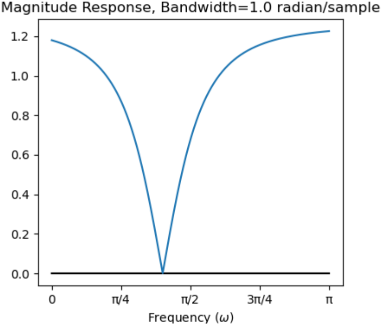
$$\frac{1}{\sqrt{2}} = \frac{|1 - e^{j(\omega - \omega_c)}|}{|1 - ae^{j(\omega - \omega_c)}|} = \frac{|1 - e^{jB/2}|}{|1 - ae^{jB/2}|} = \frac{|1 - e^{jB/2}|}{|1 - e^{\ln(a) + jB/2}|}.$$

Let's use the approximation $e^x \approx 1 + x$, and then solve for B. We get:

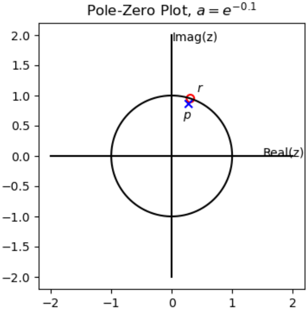
$$\frac{1}{\sqrt{2}} = \frac{|-jB/2|}{|-\ln(a) - jB/2|} \quad \Rightarrow \quad B = -2\ln(a)$$

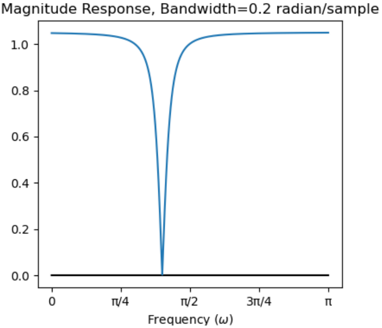
Bandwidth $B = -2\ln(a)$





Bandwidth $B = -2\ln(a)$





First-Order Notch Filter has Complex Outputs

A notch filter is

$$H(z) = \frac{1 - rz^{-1}}{1 - pz^{-1}}$$

which we implement using just one line of python:

$$y[n] = x[n] - rx[n-1] + py[n-1]$$

The problem: r and p are both complex, therefore, even if $x[n]$ is real, $y[n]$ will be complex.

Real-Valued Coefficients ⇔ Conjugate Zeros and Poles

To get real-valued coefficients, we have to use a second-order filter with complex conjugate poles and zeros ($r_2 = r_1^* = e^{-j\omega_c}$ and $p_2 = p_1^* = ae^{-j\omega_c}$):

$$\begin{aligned}
H(z) &= \frac{(1 - r_1 z^{-1})(1 - r_1^* z^{-1})}{(1 - p_1 z^{-1})(1 - p_1^* z^{-1})} \\
&= \frac{1 - (r_1 + r_1^*)z^{-1} + |r_1|^2 z^{-2}}{1 - (p_1 + p_1^*)z^{-1} + |p_1|^2 z^{-2}} \\
&= \frac{1 - 2\cos\omega_c z^{-1} + z^{-2}}{1 - 2a\cos\omega_c z^{-1} + a^2 z^{-2}}
\end{aligned}$$

So then, we can implement it as a second-order difference equation, using just one line of code in python:

$$y[n] = x[n] - 2\cos\omega_c x[n-1] + x[n-2] + 2a\cos\omega_c y[n-1] - a^2 y[n-2]$$

Real-Valued Coefficients ⇔ Conjugate Zeros and Poles

If the poles and zeros come in conjugate pairs, then we get

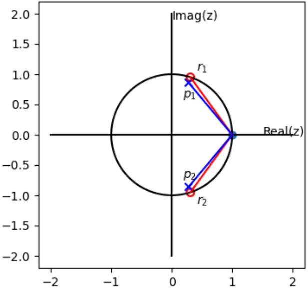
$$H(z) = \frac{(1 - r_1 z^{-1})(1 - r_1^* z^{-1})}{(1 - p_1 z^{-1})(1 - p_1^* z^{-1})} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2}}$$

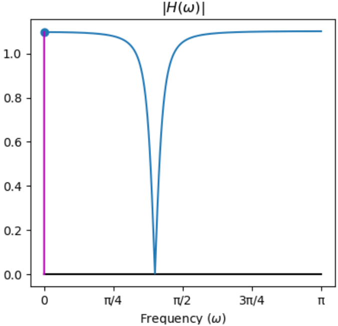
where all the coefficients are real-valued:

$$\begin{aligned} b_0 &= 1 \\ b_1 &= -2\cos\omega_c \\ b_2 &= 1 \\ a_1 &= 2a\cos\omega_c \\ a_2 &= -a^2 \end{aligned}$$

Notch Filter with Conjugate-Pair Zeros and Poles

$$|H(\omega)| = \frac{|e^{j\omega} - r_1| \times |e^{j\omega} - r_2|}{|e^{j\omega} - p_1| \times |e^{j\omega} - p_2|}$$





Summary: How to Implement a Notch Filter

To implement a notch filter at frequency ω_c radians/sample, with a bandwidth of $-\ln(a)$ radians/sample, you implement the difference equation:

$$y[n] = x[n] - 2\cos(\omega_c)x[n-1] + x[n-2] + 2a\cos(\omega_c)y[n-1] - a^2 y[n-2]$$

which gives you the notch filter

$$H(z) = \frac{(1 - r_1 z^{-1})(1 - r_1^* z^{-1})}{(1 - p_1 z^{-1})(1 - p_1^* z^{-1})}$$

Try the quiz!

Try the quiz!

Outline

Summary: How to Implement a Notch Filter

To implement a notch filter at frequency ω_c radians/sample, with a bandwidth of $-\ln(a)$ radians/sample, you implement the difference equation:

$$y[n] = x[n] - 2\cos(\omega_c)x[n-1] + x[n-2] + 2a\cos(\omega_c)y[n-1] - a^2 y[n-2]$$

which gives you the notch filter

$$H(z) = \frac{(1 - r_1 z^{-1})(1 - r_1^* z^{-1})}{(1 - p_1 z^{-1})(1 - p_1^* z^{-1})}$$

with the magnitude response:

$$|H(\omega)| = \begin{cases} 0 & \omega_c \\ \frac{1}{\sqrt{2}} & \omega_c \pm \ln(a) \\ \approx 1 & \omega < \omega + \ln(a) \text{ or } \omega > \omega - \ln(a) \end{cases}$$