
Ch. 7: Z-transform

- Definition
- Properties
 - linearity / superposition
 - time delay
 - convolution: $y[n] = h[n] * x[n] \iff Y(z) = H(z) X(z)$
- Inverse z-transform by coefficient matching
- System function $H(z)$
 - poles, zeros, pole-zero plots
 - conjugate pairs
 - relationship to $\mathcal{H}(\hat{\omega})$
- Cascading systems
- Filter design

Reading

- Text Ch. 7
- (Section 7.9 optional)

Summary:

$$\text{time domain view: } x[n] \rightarrow \boxed{h[n]} \rightarrow y[n] = h[n] * x[n]$$

$$\text{z-domain view: } X(z) \rightarrow \boxed{H(z)} \rightarrow Y(z) = H(z) X(z)$$

This is called the **convolution property** of the z-transform.

Rational system functions (ratios of polynomials)

$$H(z) = \frac{B(z)}{A(z)} = g \frac{\prod_i (z - z_i)}{\prod_j (z - p_j)} = g \frac{(z - z_1)(z - z_2) \cdots (z - z_M)}{(z - p_1)(z - p_2) \cdots (z - p_N)},$$

where $B(z)$ is a M th-order polynomial with M roots $\{z_i, i = 1, \dots, M\}$ called the **zeros** of $H(z)$, and $A(z)$ is a N th-order polynomial with N roots $\{p_j, j = 1, \dots, N\}$ called the **poles** of $H(z)$.

Graphical representation: **pole-zero plot**

Relation between **frequency response** and **system function**

$$\boxed{\mathcal{H}(\hat{\omega}) = H(z) \Big|_{z=e^{j\hat{\omega}}} = H(e^{j\hat{\omega}})}$$

System functions of causal FIR filters

$$h[n] = \sum_{k=0}^M b_k \delta[n - k] \iff H(z) = \sum_{k=0}^M b_k z^{-k}$$

Rewriting in two forms:

$$H(z) = \frac{b_0 z^M + b_1 z^{M-1} + \cdots + b_{M-1} z + b_M}{z^M} = \frac{(z - z_1)(z - z_2) \cdots (z - z_M)}{z^M}$$

where the z_k 's denote the M roots of the numerator polynomial, *i.e.*, the M zeros of the system. Use MATLAB's `roots` function.

Fact. A M th-order causal FIR filter has M zeros and has M poles at $z = 0$.

We consider only the usual case of real filters (so the filter coefficients, the b_k 's, are real numbers).

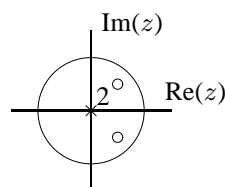
The by the **Fundamental theorem of algebra**, all of the roots of the numerator polynomial are either real, or come in complex conjugate pairs.

Fact. The zeros (and poles) of a real system are either real or come in complex conjugate pairs.

Example. The system function for a FIR filter with impulse response $h[n] = \delta[n] - \delta[n - 1] + \frac{1}{2}\delta[n - 2]$ is:

$$H(z) = 1 - z^{-1} + \frac{1}{2}z^{-2} = \frac{z^2 - z + \frac{1}{2}}{z^2} = \frac{[z - \frac{1}{2}(1 + j)][z - \frac{1}{2}(1 - j)]}{z^2},$$

because the roots of the numerator polynomial $Az^2 + Bz + C$ are at $z_{\pm} = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} = \frac{1}{2}(1 \pm j)$.



So the pole-zero diagram is:

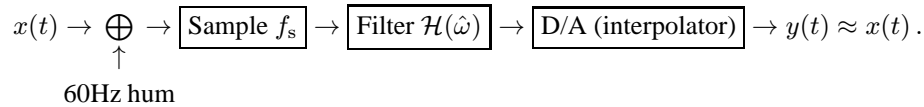
The zeros are in complex-conjugate pairs.

What about noncausal FIR filters? (Number of poles at the origin varies.)

First FIR filter design attempt: 60Hz notch filter.

Equipped with the concepts developed thus far, we can finally attempt our first **filter design!**

In many applications, such as measuring brain signals, “60Hz hum” from AC power lines contaminates the signal $x(t)$. Another example is home computer networks that operate over house AC wiring. To reduce the resulting interference, often we need to pass the signal through a filter that removes the 60Hz component while leaving other frequency components relatively unaffected. Such filtering can be done using RLC circuits, or by applying digital signal processing.



For simplicity of the algebra, we suppose that the sampling rate is $f_s = 480\text{Hz}$.

What is the corresponding digital frequency? $\hat{\omega} = 2\pi f/f_s = 2\pi 60/480 = \pi/4$.

What is the ideal frequency response $\mathcal{H}(\hat{\omega})$?

(**Picture**) of ideal $\mathcal{H}(\hat{\omega})$ with zeros at $\pm\pi/4$.

This is called a **notch filter**.

Design 1

Where do we put zeros in the pole-zero diagram to eliminate the frequency component $\hat{\omega} = \pm\pi/4$?

See pole-zero diagram below with two zeros at $\pm e^{j\pi/4}$.

Where do the poles go? Two at origin needed for causal FIR filter.

This system design looks promising, since it will certainly eliminate the 60Hz ($\hat{\omega} = \pm\pi/4$) frequency component completely.

Next we find the **impulse response** (needed for implementation) and the **frequency response** (to see if we met our design goal).

Choosing gain=1 for now, we see from the pole-zero plot:

$$H(z) = \frac{[z - e^{j\pi/4}][z - e^{-j\pi/4}]}{z^2} = \frac{z^2 - z2 \cos(\pi/4) + 1}{z^2} = 1 - z^{-1}2 \cos(\pi/4) + z^{-2} = 1 - z^{-1}\sqrt{2} + z^{-2}.$$

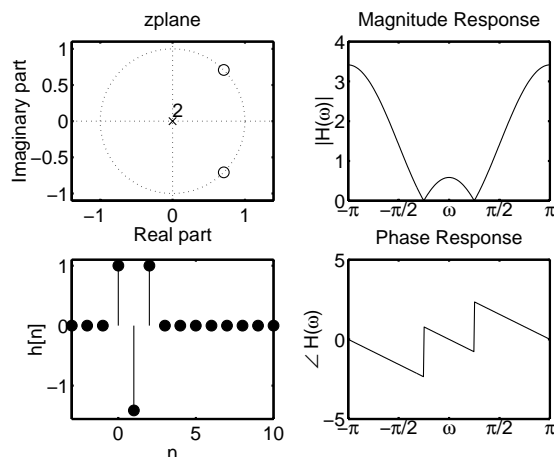
Thus the impulse response is: $h[n] = \delta[n] - \sqrt{2}\delta[n-1] + \delta[n-2]$. This is a very simple filter! It can be implemented using the following simple diffeq:

$$y[n] = x[n] - \sqrt{2}x[n-1] + x[n-2].$$

The frequency response of this FIR filter is

$$\mathcal{H}(\hat{\omega}) = 1 - \sqrt{2}e^{-j\hat{\omega}} + e^{-j2\hat{\omega}} = e^{-j\hat{\omega}}(2 \cos \hat{\omega} - \sqrt{2}).$$

Here is a plot of the **magnitude response** and **phase response** of this system, computed using MATLAB's `zplane` and `freqz` functions.



Design 2

Our first design does remove the $\hat{\omega} = \pm\pi/4$ frequency component completely, but unfortunately it also has the side effect of amplifying the high frequencies relative to the low frequencies.

How can we attenuate the high frequencies somewhat?

From the general expression for $H(z)$ above, the magnitude response is

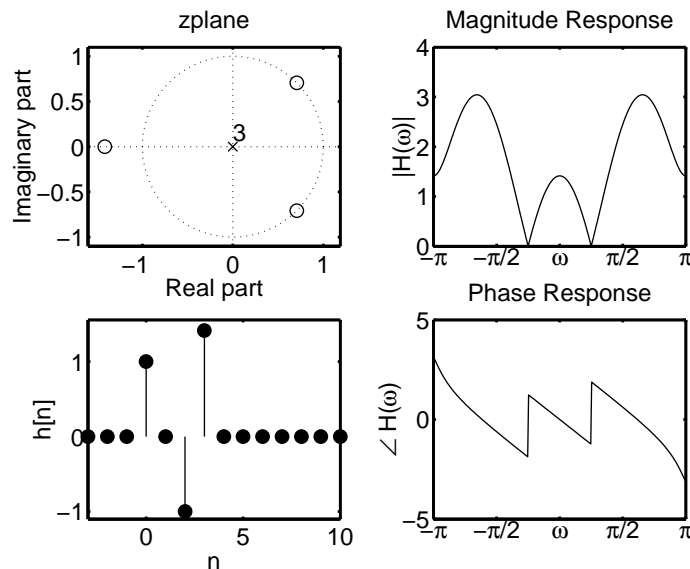
$$|\mathcal{H}(\hat{\omega})| = |H(e^{j\hat{\omega}})| = \left| \frac{(e^{j\hat{\omega}} - z_1)(e^{j\hat{\omega}} - z_2) \cdots (e^{j\hat{\omega}} - z_M)}{(e^{j\hat{\omega}})^M} \right| = |e^{j\hat{\omega}} - z_1| |e^{j\hat{\omega}} - z_2| \cdots |e^{j\hat{\omega}} - z_M|.$$

So the magnitude response is simply the product of the distances from the point $e^{j\hat{\omega}}$ (on the unit circle) to all the zeros. In particular, whenever $e^{j\hat{\omega}}$ is *close* to a zero of the system function, the magnitude response will be reduced.

So to reduce (but not eliminate) high frequency components ($\hat{\omega} \approx \pm\pi$), we can place a zero somewhere near $z = e^{j\pi} = -1$.

Comparing $\mathcal{H}(0) = H(1)$ to $\mathcal{H}(\pi) = H(-1)$, one can show that a zero at $z = -\sqrt{2}$ will equalize the gain at $\hat{\omega} = 0$ and $\hat{\omega} = \pi$.

Here is the resulting frequency response.



Well, we have equalized the gain at $\hat{\omega} = 0$ and $\hat{\omega} = \pi$, but we are still far from our ideal frequency response!

What is the output of this filter if the input signal is $x[n] = 3 + \cos(\frac{\pi}{4}n) + 7(-1)^n$?

Recall that

$$x[n] = \cos(\hat{\omega}n + \phi) \rightarrow \boxed{\mathcal{H}(\hat{\omega})} \rightarrow y[n] = |\mathcal{H}(\hat{\omega})| \cos(\hat{\omega}n + \phi + \angle\mathcal{H}(\hat{\omega})).$$

For the system above,

$$H(z) = \frac{[z - e^{j\pi/4}][z - e^{-j\pi/4}][z + \sqrt{2}]}{z^3} = [1 - z^{-1}\sqrt{2} + z^{-2}][1 + \sqrt{2}z^{-1}]$$

$$\mathcal{H}(\hat{\omega}) = H(e^{j\hat{\omega}}) = [1 - \sqrt{2}e^{-j\hat{\omega}} + e^{-j2\hat{\omega}}][1 + \sqrt{2}e^{-j\hat{\omega}}].$$

The frequencies of the input signal $x[n]$ above are $\hat{\omega} = 0$, $\hat{\omega} = \pi/4$, and $\hat{\omega} = \pi$.

Substituting into $\mathcal{H}(\hat{\omega})$ we have $\mathcal{H}(0) = (2 - \sqrt{2})(1 + \sqrt{2}) = \sqrt{2}$, $\mathcal{H}(\pi/4) = 0$ (due to zeros at $\pm\pi/4$ on unit circle), $\mathcal{H}(\pi) = (2 + \sqrt{2})(1 - \sqrt{2}) = -\sqrt{2} = \sqrt{2}e^{j\pi}$.

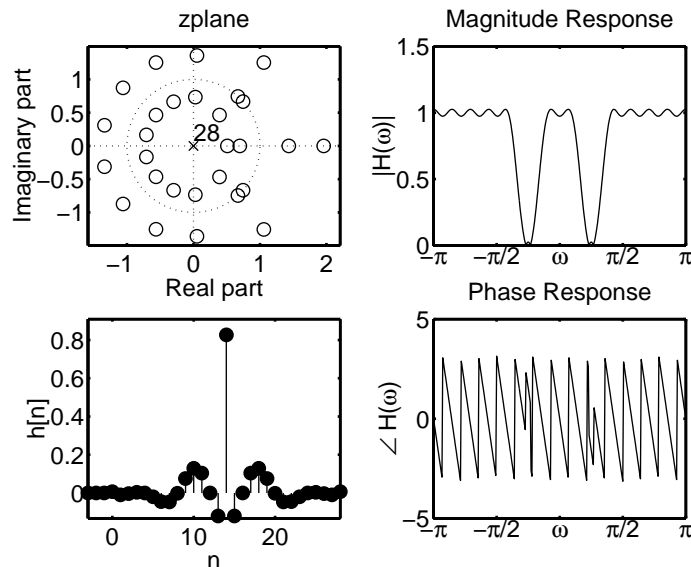
Thus the output signal is

$$y[n] = 3\sqrt{2} + 0 + \sqrt{2}\cos(\pi n + \pi).$$

A magnitude response plot shows the response to an *eternal* sinusoid.
 What about “suddenly turned on” signals $\cos(\hat{\omega}n)u[n]$, (e.g., after system first started)?
 After short initial **transient response**, the output signal approaches the **steady state** response.

Design 3

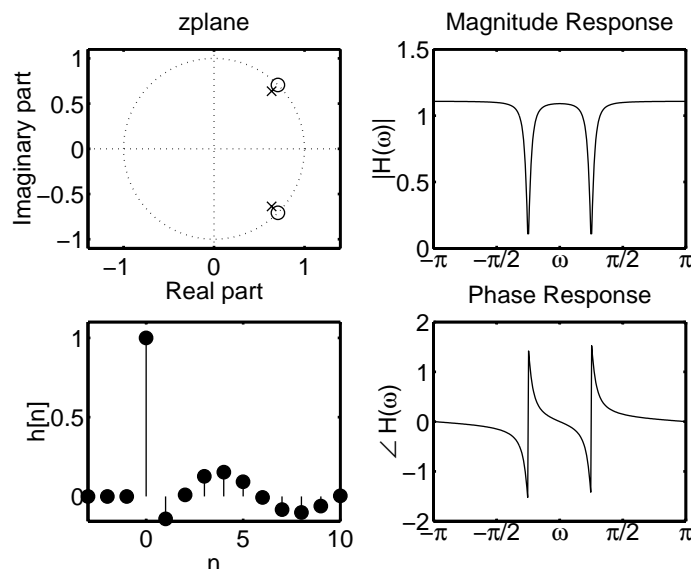
How well can we do with a more sophisticated FIR filter?
 Here is a $M = 28$ filter designed by MATLAB’s `remez` function.
 The frequency response plots were made using MATLAB’s `freqz` function.
 The impulse response plots were made using MATLAB’s `filter` function.



This is better since it has reasonably uniform gain in the passbands, but the “notch” may be somewhat wide.

Design 4

How can we get even closer to the ideal frequency response?
 One way is to use the filter with the following pole-zero plot and frequency response.



What is fundamentally different about this filter?

It is an **IIR filter**, since there are poles not at $z = 0$.

Even though we have not yet studied IIR filters in detail, we can use what we have learned so far to find

- the **system function** $H(z)$,
- the **diffeq**,
- and the **frequency response**.

However, we do not yet know how to find the **impulse response** of an IIR filter. That comes in the next chapter.

Assuming that the gain=1, from the pole-zero we have that the **system function** is:

$$H(z) = \frac{[z - e^{j\pi/4}][z - e^{-j\pi/4}]}{[z - re^{j\pi/4}][z - re^{-j\pi/4}]} = \frac{z^2 - z2 \cos(\pi/4) + 1}{z^2 - z2r \cos(\pi/4) + r^2} = \frac{1 - z^{-1}2 \cos(\pi/4) + z^{-2}}{1 - z^{-1}2r \cos(\pi/4) + r^2 z^{-2}} = \frac{Y(z)}{X(z)},$$

where r denotes the distance of the pole from the origin.

To find the **diffeq**, we first **cross multiply** as follows:

$$Y(z) [1 - z^{-1}2r \cos(\pi/4) + r^2 z^{-2}] = X(z) [1 - z^{-1}2 \cos(\pi/4) + z^{-2}].$$

Now (using linearity and shift property of z-transform), we convert back to the time domain:

$$y[n] - 2r \cos(\pi/4)y[n-1] + r^2 y[n-2] = x[n] - 2 \cos(\pi/4)x[n-1] + x[n-2].$$

Rearranging yields the following **recursive** expression:

$$y[n] = 2r \cos(\pi/4)y[n-1] - r^2 y[n-2] + x[n] - 2 \cos(\pi/4)x[n-1] + x[n-2].$$

The output at time n depends on the two past output values *and* on the current input and 2 past input values. This is very easy to implement! Much easier than a 28-tap FIR filter.

To understand the **frequency response** qualitatively, we return to the first **factored form** of the system function above:

$$\mathcal{H}(\hat{\omega}) = H(e^{j\hat{\omega}}) = \frac{[e^{j\hat{\omega}} - e^{j\pi/4}][e^{j\hat{\omega}} - e^{-j\pi/4}]}{[e^{j\hat{\omega}} - re^{j\pi/4}][e^{j\hat{\omega}} - re^{-j\pi/4}]}.$$

In particular, we find the **magnitude response** by taking the magnitude of both sides:

$$|\mathcal{H}(\hat{\omega})| = \frac{|e^{j\hat{\omega}} - e^{j\pi/4}| |e^{j\hat{\omega}} - e^{-j\pi/4}|}{|e^{j\hat{\omega}} - re^{j\pi/4}| |e^{j\hat{\omega}} - re^{-j\pi/4}|}.$$

In words, this means that for any frequency $\hat{\omega}$, the magnitude response is the product of all the distances from the point $e^{j\hat{\omega}}$ on the unit circle to all the zeros, divided by the product of all the distances from the point $e^{j\hat{\omega}}$ to all the poles.

Being close to a zero decreases the magnitude response; being close to pole increases the magnitude response.

For any frequency $\hat{\omega}$ that is far from $e^{j\pi/4}$ in the notch filter example, the distance from $e^{j\hat{\omega}}$ to the zero and to its neighboring pole is almost the same, so the ratio is nearly unity and thus the magnitude response is nearly unity. The zero is slightly further away, so the magnitude response is slightly more than unity.

How could we make the response closer to unity far from the zero?

Move the pole closer to the zero. (But, as we will see in the next chapter, this makes the transient response “longer,” *i.e.*, the impulse response decays to zero slower, which can be undesirable in some applications.)

Interconnection of LTI systems

Engineers build complicated systems by connecting simpler components.

There are three principal ways of connecting two LTI systems: series, parallel, and in a feedback loop.

Series connection

Here are two LTI systems connected in **series** or in **cascade**:

$$x[n] \rightarrow \boxed{H_1(z)} \rightarrow \boxed{H_2(z)} \rightarrow y[n].$$

In the time domain, we have seen that that overall system is LTI with impulse response $h[n] = h_1[n] * h_2[n]$.

Since time-domain convolution corresponds to multiplication in the z-domain, the overall system function is

$$H(z) = H_1(z) \cdot H_2(z),$$

the *product* of the two individual system functions.

Example. Application: channel equalization for wireless communications. H_1 represents the channel (attenuation, multipath) and H_2 represents the equalizer that is designed to “undo” (invert) the distortions induced by the channel.

Example. Application: Dolby noise reduction...

Example. Putting two unit-sample delays in series is equivalent to a two-sample delay:

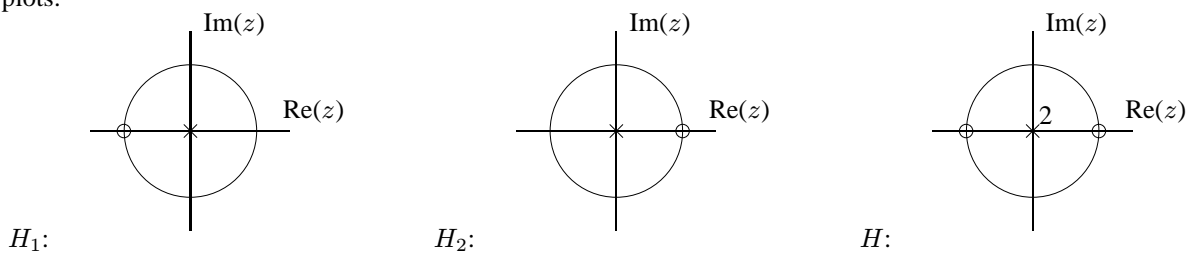
$$x[n] \rightarrow \boxed{z^{-1}} \rightarrow \boxed{z^{-1}} \rightarrow y[n] \text{ is equivalent to } x[n] \rightarrow \boxed{z^{-2}} \rightarrow y[n].$$

Example. $H_1(z) = g_1(1 + z^{-1}) = g_1 \frac{z+1}{z}$ and $H_2(z) = g_2(1 - z^{-1}) = g_2 \frac{z-1}{z}$.

In series:

$$H(z) = H_1(z)H_2(z) = g_1g_2 \frac{z+1}{z} \frac{z-1}{z} = g_1g_2 \frac{(z+1)(z-1)}{z^2}.$$

Pole-zero plots:



For **rational** system functions:

$$H(z) = H_1(z)H_2(z) = \frac{B_1(z) B_2(z)}{A_1(z) A_2(z)}.$$

So the zeros of $H(z)$ are all the zeros of the two systems connected in series, and likewise for the poles, except in cases where $B_1(z)$ and $A_2(z)$ share a common root, or where $B_2(z)$ and $A_1(z)$ share a common root, which is called **pole-zero cancellation**.

Parallel connection
(Picture)

$$H(z) = H_1(z) + H_2(z) = \frac{B_1(z)}{A_1(z)} + \frac{B_2(z)}{A_2(z)} = \frac{B_1(z)A_2(z) + B_2(z)A_1(z)}{A_1(z)A_2(z)}.$$

In this case we see that the parallel connected system has all the poles of the two component systems (unless there is pole-zero cancellation).

But we can say very little if anything about the zeros in general.

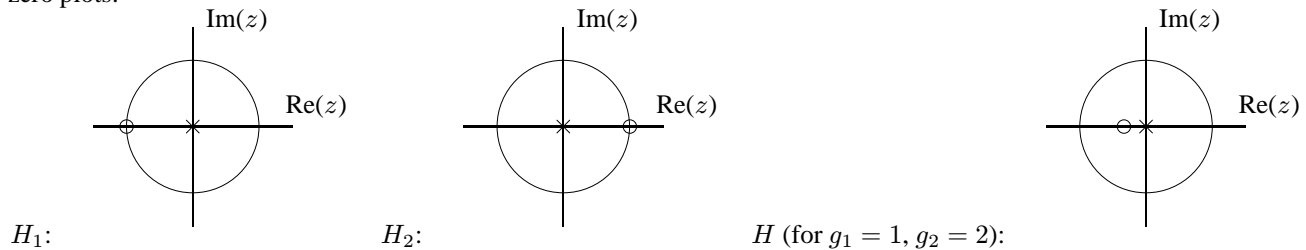
Example. $H_1(z) = g_1(1 + z^{-1}) = g_1 \frac{z+1}{z}$ and $H_2(z) = g_2(1 - z^{-1}) = g_2 \frac{z-1}{z}$.

In parallel:

$$H(z) = H_1(z) + H_2(z) = g_1 \frac{z+1}{z} + g_2 \frac{z-1}{z} = \frac{(g_1 + g_2)z + (g_2 - g_1)}{z}.$$

Single pole at $z = 0$ and zero at $z = (g_1 - g_2)/(g_1 + g_2)$.

Pole-zero plots:



Feedback connection

Example. automobile cruise control.

(Picture of system $H_1(z)$ with negative feedback $H_f(z)$)

Deriving:

$$H(z) = \frac{H_1(z)}{1 + H_1(z)H_f(z)}$$

Location of poles and zeros of $H(z)$ is complicated; indeed, part of the purpose of feedback is to stabilize and otherwise unstable system (e.g., Harriet airplane). Even if $H_1(z)$ has poles that are outside the unit circle, which means it is an unstable system, it can still be the case that $H(z)$ has all its poles inside the unit circle, meaning that it is stable, if the feedback system $H_f(z)$ is designed properly.

Example. $H_1(z) = \frac{1}{1 - az^{-1}} = \frac{z}{z - a}$. If $a > 1$, then the system is unstable.

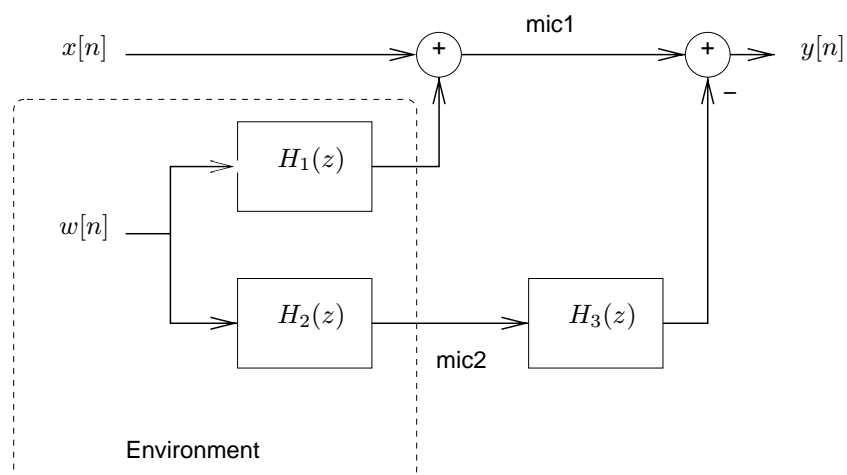
The simplest possible feedback controller would be $H_f(z) = g$ for some constant g .

$$H(z) = \frac{H_1(z)}{1 + H_1(z)H_f(z)} = \frac{\frac{z}{z-a}}{1 + \frac{z}{z-a}g} = \frac{z}{z - a + zg} = \frac{z}{z(1+g) - a},$$

so the system with feedback has a pole at $z = a/(1+g)$.

If $g > a - 1$, then the pole will be “moved” to within the unit circle, thereby stabilizing the system.

Example. Noise cancellation. (Explained in lecture)



For perfect noise cancellation, we need $H_3(z) = 1/H_2(z)$. If $H_1(z)$ is FIR, then $H_3(z)$ will be IIR, which again brings us to the next chapter..