**FIR vs. IIR**

- one wall
- one-shot delay
- feedforward

- two walls
- recirculating delay
- feedback

\[
\text{Out} = \text{in} + \text{delayedIn}
\]

\[
\text{Out} = \text{in} + \text{delayedOut}
\]
...as math

\[ x_n = \text{this input sample} \]
\[ x_{(n-N)} = \text{Nth previous input} \]

\[ y_n = a_0 x_n + a_N x_{(n-N)} \]

FIR

\[ y_{(n-N)} = \text{Nth previous output} \]

\[ y_n = a_0 x_n - b_N y_{(n-N)} \]

IIR
use a pencil

\[ x_{0,1,2,3\ldots} = 0,1,0,0,0,0,0,\ldots \]

\[ y_{n} = a_{0}x_{n} + a_{1}x_{(n-1)} \]

\[ y_{0} = 0 \]

write out \( y_{1,2,3\ldots} = \ldots \)
impulse responses

click impulse

FIR (time smear / spectral change)

IIR (damped exponential decay)
FIR vs. IIR

- one wall
- one-shot delay
- feedforward

Out = in + delayedIn

test FIR in webchuck

- two walls
- recirculating delay
- feedback

Out = in + delayedOut

test IIR in webchuck
\[ a_0 = 0.5 \]
\[ a_1 = 0.5 \]
\[ x_0 = 0 \]
\[ y_n = a_0 x_n + a_1 x_{(n-1)} \]

\[ a_0 = 0.5 \]
\[ b_1 = -0.5 \]
\[ y_0 = 0 \]
\[ y_n = a_0 x_n - b_1 y_{(n-1)} \]
the all-in-one filter difference equation (as C code)

/* This code implements the standard difference equation, (Cook: pg. 26)
y[0] = g \ast (x[0] + a[1] \ast x[1] \ldots + a[N] \ast x[N])
    - b[1] \ast y[1] - \ldots - b[M] \ast y[M]

For simplicity, N = M = "ORDER" */

Out = In + delayedIn - delayedOut

[0] = this sample
[1] = last sample
.
.
.
[N] = Nth delayed sample
convolution (FIR)

click impulse

\[ a_0, a_1 \]

(time scale from 0 to 0.002)

(impulse response = coefficients!)
convert any IR to FIR

Low order FIR
a0, a1

High order FIR
a0, a1, a2, a3, a4,...

(coefficients made from IR of IIR)
summary

- filters, linear, don't add frequencies (as opposed to modulation or distortion, non-linear)
- filters change duration and spectral weighting
- size from big to little: echo, pitch, eq, image pos.
- Sharpness ('Q') from high to low: string, tube
- it's hard to find simple filters in nature, most are combinations, complexes