Ch. 5-8: Filtering

Ch. 5: FIR filters

- Time domain analysis
- System properties:
  - linearity, time-invariance, causality, stability
- Impulse response
- Convolution

Ch. 6: Frequency domain analysis of filters (mostly FIR)

- Frequency response of filters
- Response to sinusoids and complex exponentials
- Response to periodic signals

Ch. 7: Z-transform

- Powerful “frequency domain” analysis technique
  - filter design

Ch. 8: IIR filters

- Time and frequency domain analysis
- Design using z-domain

Reading

- Text Ch. 5-8

Ch. 5. FIR Filters

Introduction

Thus far we have focused primarily on **signals**. We now turn to **systems** for the remainder of the course.

Very often we have at hand a (digital) signal $x[n]$ whose properties are somehow less than ideal, so we would like to **process** (digitally) that signal $x[n]$ to create a new signal $y[n]$ that is somehow an improved version of $x[n]$.

What we need to *design* is a **system** that accepts a signal as its **input**, usually denoted $x[n]$, and produces a modified signal as its **output**, or **response**, usually denoted $y[n]$. We illustrate this situation generally with the following diagram.

$$\begin{tikzcd} \text{Input signal} & & \text{Output signal} \\ x[n] \arrow[r] & \boxed{\text{Discrete-time System}} \arrow[r] & y[n] \end{tikzcd}$$

This is a **single-input, single-output** system, which is the principal variety that we discuss in this course.

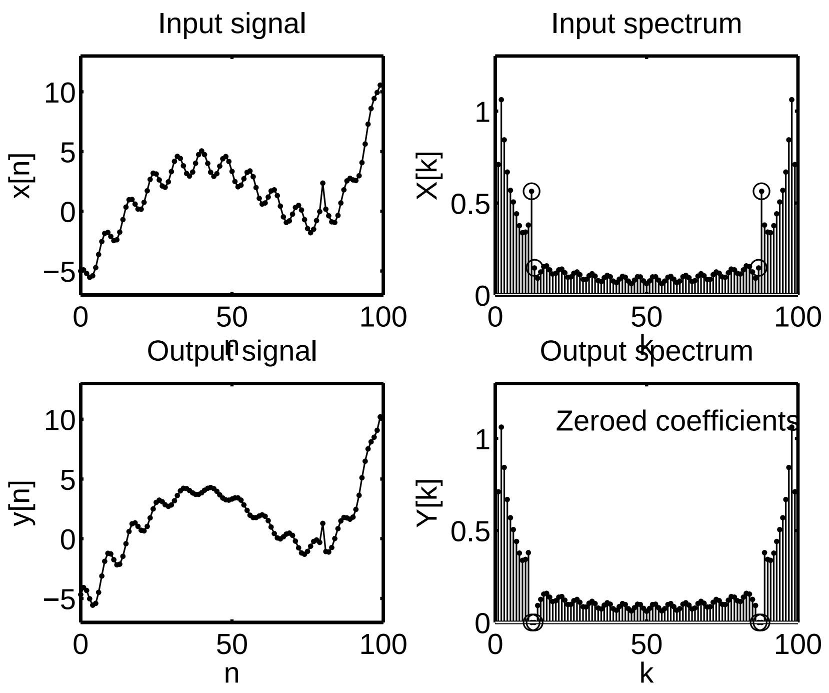
Example. Many audio playback systems have some kind of "bass boost" feature that, when engaged, takes the signal and produces a new version in which the low frequencies are amplified.

Example. Signals contaminated by 60Hz hum (a 60Hz sinusoid).

How could we remove such "hum" using principles covered thus far in the course?
Could use DFT techniques but what if 60Hz does not correspond to an integer k? Recall that $f_k = \frac{k}{N} f_s$ for $0 \le k \le f_s/2$.
For example, if $N = 100$ and $f_s = 480$, then $60 = \frac{k}{100}480$ means we need $k = 60/480 \cdot 100 = 12.5$, which is not an integer.
The following plot shows result of setting the $k = 12$ and $k = 13$ DFT values to zero, *i.e.*,

$$Y[k] = \begin{cases} 0, & k = 12, 13, 87, 88 \\ X[k] & \text{otherwise,} \end{cases}$$

and then applying the inverse DFT (the synthesis equation) for form $y[n]$. But the hum is incompletely removed.



Even if k were an integer, a DFT approach is poorly suited to **real-time** use where the input signal samples are arriving continuously and the output signal values are needed immediately, such as in a public address system. A DFT approach can be fine for "**off line**" purposes and for signals like images that are functions of an argument like spatial location rather than time.

Systems and their input-output relationships

We represent the operation of taking an **input signal** and modifying to produce an **output signal** by the following block diagram.

$$x[n] \to \boxed{\mathcal{T}} \to y[n]\,,$$

where $\mathcal{T}$ denotes the mathematical operator that defines the **discrete-time system**.

Mathematically we write

$$y = \mathcal{T}\{x\}$$

or sometimes

$$x[n] \xrightarrow{\mathcal{T}} y[n]\,.$$

The book writes "$y[n] = \mathcal{T}\{x[n]\}$" but this is dangerous since it suggests that the output at time n depends only on the input at time n, which is not true in general.

The **goal** if the rest of the course is to learn how to *design* a system $\mathcal{T}$ to achieve desired effects on signals, such as amplifying or attenuating certain frequency components. But first we must spend a long time simply *analyzing* such systems.

We use the symbol $\mathcal{T}$ when we refer to systems in general.
When we refer to a specific system, we define its **input-output relationship**.

Example. The (noncausal) 3-point moving average **filter** has the following **input-output relationship**:

$$y[n] = \frac{1}{3}\left(x[n-1] + x[n] + x[n+1]\right).$$

In words, the output signal value at any time n, considered to be the "current" time, is the *average* of the current signal value, the previous signal value, and the next signal value.

Does this filter do a good job of attenuating 60 Hz hum? Let us determine the output signal when the input is a 60 Hz sinusoid sampled at $f_{\mathrm{s}} = 480$ Hz, so

$$x[n] = \cos(2\pi\frac{1}{8}n)$$

$$\begin{aligned} y[n] &= \frac{1}{3}\left(x[n-1] + x[n] + x[n+1]\right) \\ &= \frac{1}{3}\Big(\cos(\frac{\pi}{4}(n-1)) + \cos(\frac{\pi}{4}n) + \cos(\frac{\pi}{4}(n+1))\Big) \\ &= \frac{1}{3}\Big(\cos(\frac{\pi}{4}n - \pi/4) + \cos(\frac{\pi}{4}n) + \cos(\frac{\pi}{4}n + \pi/4)\Big) \\ &= \frac{(1+\sqrt{2})}{3}\cos(\frac{\pi}{4}n) \approx 0.8\cos(\frac{\pi}{4}n) \end{aligned}$$

So a 60 Hz hum signal (for $f_{\mathrm{s}} = 480$Hz) is attenuated only by about 20%, which still leaves a lot of hum power.

We must also be able to understand the input-output relationship graphically.

Example. $x[n] = 6u[n-3]$ where we define the following **unit step function**:

$$u[n] \triangleq \begin{cases} 1, & n \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$ ***(Picture)***

Find the output $y[n]$.

Example. $x[n] = 3\delta[n-4]$ where we define the following the **unit impulse function**, aka **Kronecker impulse function**:

$$\delta[n] \triangleq \begin{cases} 1, & n = 0 \\ 0, & \text{otherwise.} \end{cases}$$ ***(Picture)***

Find the output $y[n]$.

System properties

- Time-behavior properties
  - causal vs noncausal
  - time-invariant vs time-varying
- Other properties
  - linear vs nonlinear
  - stable vs unstable
  - invertible vs non-invertible
- LTI systems
  - impulse response $h[n]$
  - length of impulse response: FIR vs IIR

Causality

Definition

A system is **causal** if the output $y[n]$ at time n depends only on the current input value $x[n]$ and past input values: $x[n-1]\,,\ x[n-2]\,, \ldots$.

Mathematically: $y[n] = F(x[n]\,, x[n-1]\,, x[n-2]\,, \ldots)$ where $F(\cdot)$ is some function.

Otherwise the system is called **noncausal**.

Causality is necessary for real-time implementation, but many DSP problems involved stored data (post-processing), *e.g.*, image processing (OCR) or restoration of analog audio recordings, so non-causal systems are also relevant.

Is the previous moving average filter causal? No.

Example. Modified **moving average** or **running average** filter:

$$y[n] = \frac{1}{3}\left(x[n] + x[n-1] + x[n-2]\right).$$

Causal? Yes. Can process signals in “real time.”

Static vs dynamic

- For a **static system** or **memoryless** system, the output $y[n]$ depends only on the current input $x[n]$, not on previous or future inputs. Example: $y[n] = x^2[n]$.
- Otherwise it is a **dynamic system** and must have memory.

Dynamic systems are the interesting ones and will be our focus. (This time we take the more complicated choice!)

Is a memoryless system necessarily causal? Yes. But dynamic systems can be causal or noncausal.

Time invariance

Systems whose input-output behavior does not change with time are called **time-invariant** and will be our focus.
Why?
- "Easier" to analyze.
- Time-invariance is a desired property of many systems.

A (relaxed) system $\mathcal{T}$ is called **time invariant** or **shift invariant** iff

$$x[n] \xrightarrow{\mathcal{T}} y[n] \quad \text{implies that} \quad x[n-n_0] \xrightarrow{\mathcal{T}} y[n-n_0]$$

for *every* input signal $x[n]$ and *integer* time shift n_0.

Otherwise the system is called **time variant** or **shift variant**.

Graphically:

$$\begin{tikzcd} x[n] \arrow[r] & \boxed{\text{system } \mathcal{T}} \arrow[r, "{y[n]}"] & \boxed{\text{delay } z^{-n_0}} \arrow[r] & y_1[n] = y[n-n_0] \\ x[n] \arrow[r] & \boxed{\text{delay } z^{-n_0}} \arrow[r, "{x_2[n]}"] & \boxed{\text{system } \mathcal{T}} \arrow[r] & y_2[n] \end{tikzcd}$$

where $x_2[n] = x[n-n_0]$

Recipe for showing time-invariance.
- Determine the output signal $y[n]$ due to a generic input $x[n]$. Delay that output to form $y_1[n] = y[n-n_0]$ for a generic shift n_0.
- Determine output signal $y_2[n]$ due to delayed input signal $x_2[n] = x[n-n_0]$.
- If $y_2[n] = y_1[n]$, then the system is time-invariant.

(If you cannot show that $y_2[n] = y_1[n]$, try to find a counter-example to demonstrate that the system is time varying.

Example: 3-point **moving average** $y[n] = \frac{1}{3}(x[n-1] + x[n] + x[n+1])$. Time invariant? yes.
- Clearly, delaying the output yields $y_1[n] = y[n-n_0] = \frac{1}{3}(x[n-n_0-1] + x[n-n_0] + x[n-n_0+1])$
- Output due to shifted input $x_2[n] = x[n-n_0]$ is

$$y_2[n] = \frac{1}{3}(x_2[n-1] + x_2[n] + x_2[n+1]) = \frac{1}{3}(x[n-n_0-1] + x[n-n_0] + x[n-n_0+1]) .$$

  Note that we did this in two steps to avoid errors!
- Since $y_1[n] = y_2[n]$, the system is time-invariant.

Example: down-sampler $y[n] = x[2n]$. Time invariant? no. How do we show lack of a property? Find counter-example. If $x[n] = \delta[n]$ then $y[n] = \delta[n]$. If $x_2[n] = \delta[n-1]$ then $y_2[n] = 0 \neq y_1[n] = y[n-1] = \delta[n-1]$. Simple counterexample all that is needed.

We will focus mostly on time-invariant systems hereafter.

"Amplitude" properties (linearity, stability, invertibility)

Linearity

We will also focus on **linear systems**.

Why linearity?

- The class of **linear systems** is easier to analyze.
- Often linearity is desirable - avoids distortions.
- Many nonlinear systems are approximately linear, so first-order analysis is linear case.

A system $\mathcal{T}$ is **linear** iff

$$x_1[n] \xrightarrow{\mathcal{T}} y_1[n]\,, \quad x_2[n] \xrightarrow{\mathcal{T}} y_2[n] \quad \Rightarrow \quad \alpha_1 x_1[n] + \alpha_2 x_2[n] \xrightarrow{\mathcal{T}} \alpha_1 y_1[n] + \alpha_2 y_2[n]$$

for any signals $x_1[n]$, $x_2[n]$, and constants α_1 and α_2.
Otherwise the system is called **nonlinear**.

Here is a block-diagram representation of this property.

$$\begin{tikzcd}
x_1[n] \arrow[r] & \otimes \arrow[dr] & & & \\
 & \alpha_1 \arrow[u] & \bigoplus \arrow[r] & \boxed{\mathcal{T}} \arrow[r] & y[n] \\
x_2[n] \arrow[r] & \otimes \arrow[ur] & & & \\
 & \alpha_2 \arrow[u] & & &
\end{tikzcd}$$

$$\begin{tikzcd}
x_1[n] \arrow[r] & \boxed{\mathcal{T}} \arrow[r] & y_1[n] \arrow[r] & \otimes \arrow[dr] & & \\
 & & & \alpha_1 \arrow[u] & \bigoplus \arrow[r] & y[n] \\
x_2[n] \arrow[r] & \boxed{\mathcal{T}} \arrow[r] & y_2[n] \arrow[r] & \otimes \arrow[ur] & & \\
 & & & \alpha_2 \arrow[u] & &
\end{tikzcd}$$

Two important special cases of linearity property.

- **scaling property**: $\boxed{\mathcal{T}[ax[n]] = aT[x[n]]}$
  Note that from $a = 0$ we see that zero input signal implies zero output signal for a linear system.
- **additivity property**: $\boxed{\mathcal{T}[x_1[n] + x_2[n]] = \mathcal{T}[x_1[n]] + \mathcal{T}[x_2[n]]}$
  Using proof-by-induction, one can easily extend this property to the general **superposition property**:

$$\boxed{\mathcal{T}\left[\sum_{k=1}^{K} x_k[n]\right] = \sum_{k=1}^{K} \mathcal{T}[x_k[n]].}$$

In words: the response of a linear system to the sum of several signals is the sum of the response to each of the signals.
In general superposition need not hold for infinite sums; additional continuity assumptions are required.
We assume the superposition summation holds even for **infinite sums** without further comment in this course.

Example. Show that the accumulator is a linear system, where $y[n] = \sum_{k=-\infty}^{n} x[k]$.
Method:

- Find output signal $y_1[n]$ for a general input signal $x_1[n]$.
- "Repeat" for input $x_2[n]$ and $y_2[n]$.
- Find output signal $y[n]$ when input signal is $x[n] = \alpha_1 x_1[n] + \alpha_2 x_2[n]$.
- If $y[n] = \alpha_1 y_1[n] + \alpha_2 y_2[n] \ \forall n$, then the system is linear.

For the accumulator, $y_1[n] = \sum_{k=-\infty}^{n} x_1[k]$ and $y_2[n] = \sum_{k=-\infty}^{n} x_2[k]$. If the input is $x[n] = \alpha_1 x_1[n] + \alpha_2 x_2[n]$, then the output is

$$y[n] = \sum_{k=-\infty}^{n} x[k] = \sum_{k=-\infty}^{n} (\alpha_1 x_1[k] + \alpha_2 x_2[k]) = \alpha_1 \sum_{k=-\infty}^{n} x_1[k] + \alpha_2 \sum_{k=-\infty}^{n} x_2[k] = \alpha_1 y_1[n] + \alpha_2 y_2[n] \, .$$

Since this holds for all n, for all input signals $x_1[n]$ and $x_2[n]$, and for any constants α_1 and α_2, the accumulator is linear.

Example: To show that $y[n] = x^3[n]$ is nonlinear, all that is needed is a counter-example to the above properties. The scaling property will *usually* suffice[1]. Let $x_1[n] = 2$, a constant signal. Then $y_1[n] = 2^3 = 8$. Now suppose instead that the input is $x_2[n] = 10x_1[n] = 20$, then the output is $y_2[n] = 20^3 = 8000 \neq 10y_1[n] = 80$, so the system is nonlinear.

Stability

A system is **bounded-input bounded-output (BIBO) stable** iff every bounded input produces a bounded output.

$$\text{If } \exists M_x \text{ s.t. } |x[n]| \leq M_x < \infty \ \forall n, \text{ then there must exist an } M_y \text{ s.t. } |y[n]| \leq M_y < \infty \ \forall n.$$

Usually M_y will depend on M_x.

Example. An **accumulator** system: $y[n] = y[n-1] + x[n]$.
Consider input signal $x[n] = u[n]$, which is bounded by $M_x = 1$. Assuming $y[-1] = 0$, the output signal is $y[n] = (n+1)u[n]$, which increases without bound. So the accumulator is an **unstable** system.

We will derive a simple test for BIBO stability shortly.

Invertibility skip

A system $\mathcal{T}$ is called **invertible** if there exists a system $\mathcal{T}^{-1}$ that can process the output of system $\mathcal{T}$ and yield the input to $\mathcal{T}$, *i.e.*,

$$x[n] \rightarrow \boxed{\mathcal{T}} \rightarrow \boxed{\mathcal{T}^{-1}} \rightarrow x[n] \, ,$$

for any possible input signal.

Example. The system $x[n] \xrightarrow{\mathcal{T}} y[n] = 3x[n-1]$ is invertible. The inverse system is given by $y[n] \xrightarrow{\mathcal{T}^{-1}} x[n] = \frac{1}{3} y[n+1]$.

Example. The squaring system $y[n] = x^2[n]$ is not invertible since given $y[n]$, we cannot determine the sign of $x[n]$.

Example. Dolby noise reduction.

[1] In fact, it is challenging to find a system that satisfies the scaling property but is nonlinear. They do exist though. Try to find one as a challenge...

Linear filters

Having introduced important system properties, we now turn to the family of discrete-time systems that will be our focus for the rest of this chapter, called **filters**. Usually when we speak of "filters," we mean a specific type of linear[2] time-invariant (LTI) system.

For an FIR filter, the general **input-output relationship**, also called a **difference equation** here, is as follows:

$$y[n] = \sum_{k=M_1}^{M_2} b_k x[n-k] \,.$$

- $M_2 - M_1$ is called the **order** of the filter.
- The b_k's are called the **filter coefficients**.
- In this equation, the index "k" is in the *time domain*.

Properties of FIR filters

- An FIR filter is, by construction, **linear** and **time invariant**, which we abbreviate by **LTI**.
- For the system to be **causal** we must have $M_1 \geq 0$. Usually we choose $M_1 = 0$ and write:

  $$y[n] = \sum_{k=0}^{M} b_k x[n-k] = b_0 x[n] + b_1 x[n-1] + \cdots + b_M x[n-K] \,,$$

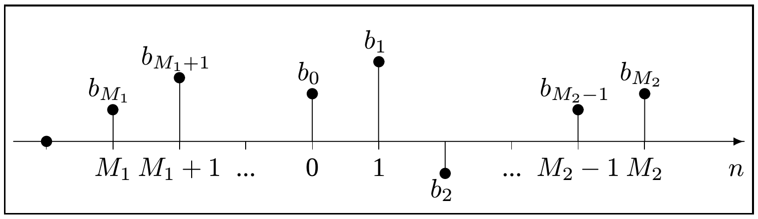
  where M is called the **order** of the filter.
- For an LTI system, the **impulse response** is the output of the system when input signal is a unit impulse signal $x[n] = \delta[n]$. We usually use $h[n]$ to denote an impulse response function. In other words,

  $$x[n] \xrightarrow{\mathcal{T}} y[n] \quad \Rightarrow \quad \delta[n] \xrightarrow{\mathcal{T}} h[n] \,.$$

  If we are given the input-output relationship for a system, then it is trivial to find the impulse response: just replace $x[n]$ by $\delta[n]$, and $y[n]$ by $h[n]$. (Later we will see more complicated cases.)

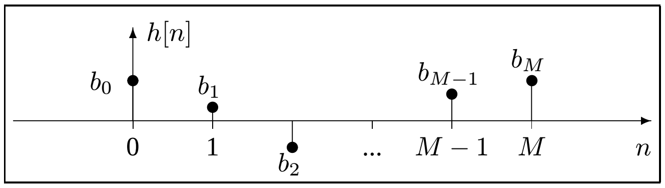
  For the above general FIR filter, the impulse response is

  $$h[n] = \sum_{k=M_1}^{M_2} b_k \delta[n-k] = \begin{cases} 0, & n < M_1 \\ b_{M_1}, & n = M_1 \\ \vdots & \\ b_{M_2}, & n = M_2 \\ 0, & n > M_2. \end{cases}$$



  - **FIR** (M_1 and M_2 both finite) chapter 5-6
  - **IIR** (M_1 or M_2 infinite) chapter 7-8
    Example. $y[n] = \frac{1}{2}y[n-1] + x[n] \Rightarrow h[n] = \sum_{k=0}^{\infty}(\frac{1}{2})^k \delta[n-k]$
- If an FIR filter is causal, what can we say about its impulse response $h[n]$? It is zero for $n < 0$ since

  $$h[n] = \sum_{k=0}^{M} b_k \delta[n-k] = \begin{cases} 0, & n < 0 \\ b_0, & n = 0 \\ b_1, & n = 1 \\ \vdots & \\ b_M, & n = M \\ 0, & n > M. \end{cases}$$



[2]There are some exceptions to this terminology. For example, in the lab you will use a **median filter** which is a nonlinear, but time-invariant, discrete-time system. But *usually* when people say "filter" then mean an LTI system.

Example. Consider the following FIR filter.

$$y[n] = \frac{1}{2-\sqrt{2}}x[n] - \frac{\sqrt{2}}{2-\sqrt{2}}x[n-1] + \frac{1}{2-\sqrt{2}}x[n-2] \approx 1.7x[n] - 2.4x[n-1] + 1.7x[n-2]\,.$$

What are the coefficients of this filter? We see: $M = 2$, $b_0 = \frac{1}{2-\sqrt{2}}$, $b_1 = \frac{\sqrt{2}}{2-\sqrt{2}}$, and $b_2 = \frac{1}{2-\sqrt{2}}$.

Is this filter a causal system? Yes.

What is the impulse response of this filter?

$$h[n] = \frac{1}{2-\sqrt{2}}\delta[n] - \frac{\sqrt{2}}{2-\sqrt{2}}\delta[n-1] + \frac{1}{2-\sqrt{2}}\delta[n-2] \approx 1.7\delta[n] - 2.4\delta[n-1] + 1.7\delta[n-2]\,.$$

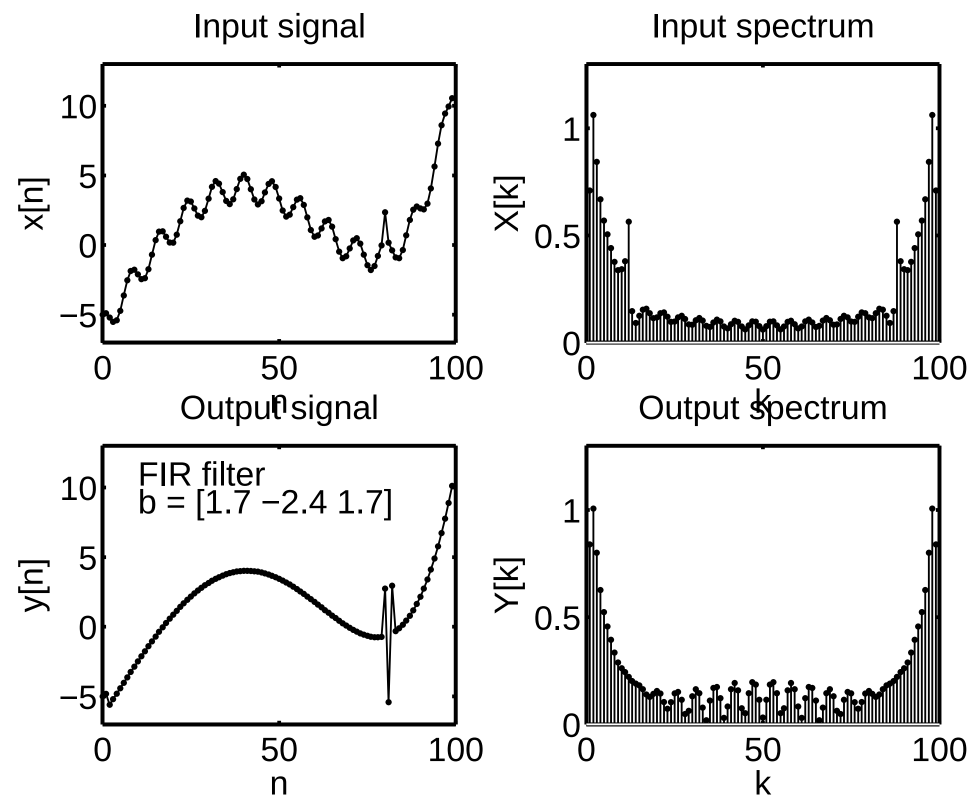
(Picture)

How well does this particular system work for removing 60 Hz hum (when $f_s = 480$ Hz)?

For the particular signal under consideration earlier, we can find out by filtering the signal using the following MATLAB command.

```
y = filter([1.7 -2.4 1.7], [1], x)
```

The first argument is the filter coefficients, the b_k's as a vector: $[b_0\ b_1\ \ldots b_M]$.



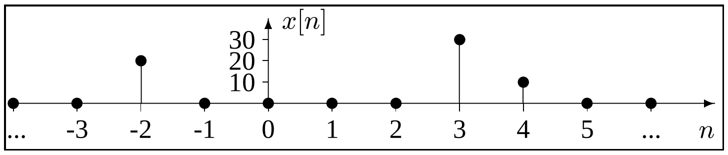
So this FIR filter appears to remove 60Hz hum pretty well.

Where did the coefficients, the b_k's, come from? This is the subject of filter design, which we will tackle in Chapter 7.

Convolution

In our discussion of Fourier series and the DFT, we said it was useful to take "complicated" signals (like sawtooth waves) and express them as the sum of "simpler" signals such as sinusoids. Sinusoids are not the only possible choice. For discrete-time signals, it is also useful to express signals as a sum of impulse functions.

Example. Consider the following signal.



The above description is graphical. We also need mathematical descriptions of such signals. One not-so-convenient mathematical formula would be

$$x[n] = \begin{cases} 20, & n = -2 \\ 30, & n = 3 \\ 10, & n = 4 \\ 0, & \text{otherwise.} \end{cases}$$

A more convenient formula is to express $x[n]$ as the sum of four impulse functions

$$x[n] = 20\delta[n+2] + 30\delta[n-3] + 10\delta[n-4]\,.$$

Why is this more convenient?
Suppose we are using an LTI system with impulse response $h[n] = 5nu[n-4]$, and we want to determine the response (the output) of that system when the above signal is the input.

Notice that we have *not* been given an input-output relationship! So how can we find $y[n]$? We have been given the following two key facts:

- The system $\mathcal{T}$ is LTI.
- The impulse response of the system $\mathcal{T}$ is $h[n] = 5nu[n-4]$, *i.e.*, $\delta[n] \xrightarrow{\mathcal{T}} h[n] = 5nu[n-4]$

So using **time-invariance**, we know that

$$\begin{aligned} \delta[n+2] &\xrightarrow{\mathcal{T}} h[n+2] = 5(n+2)u[n-2] \\ \delta[n-3] &\xrightarrow{\mathcal{T}} h[n-3] = 5(n-3)u[n-7] \\ \delta[n-4] &\xrightarrow{\mathcal{T}} h[n-4] = 5(n-4)u[n-8]\,. \end{aligned}$$

Therefore, using **linearity**, we know that

$$\begin{aligned} x[n] = 20\delta[n+2] + 30\delta[n-3] + 10\delta[n-4] \xrightarrow{\mathcal{T}} y[n] &= 20h[n+2] + 30h[n-3] + 10h[n-4] \\ &= 100(n+2)u[n-2] + 150(n-3)u[n-7] + 50(n-4)u[n-8]\,. \end{aligned}$$

So we have determined the output signal $y[n]$ for this given input signal $x[n]$ *without knowing the input-output relationship*! We could have done this for *any* possible input signal.

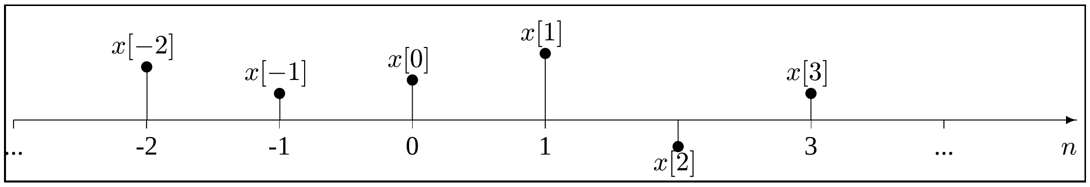
The **convolution sum**, derived next, generalizes the idea illustrated in this example.

Representing $x[n]$ using impulse functions

We can express any discrete-time signal $x[n]$ using impulse functions as follows:

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\,\delta[n-k] = \cdots x[-2]\,\delta[n+2] + x[-1]\,\delta[n+1] + x[0]\,\delta[n] + x[1]\,\delta[n-1] + x[2]\,\delta[n-2] + \cdots .$$

Example.



Given an LTI system with impulse response $h[n]$, using **time invariance** we know that

$$\delta[n] \xrightarrow{\mathcal{T}} h[n] \quad \Rightarrow \quad \delta[n-k] \xrightarrow{\mathcal{T}} h[n-k] .$$

Using the scaling property associated with **linearity**, we know that

$$x[k]\,\delta[n-k] \xrightarrow{\mathcal{T}} x[k]\,h[n-k] .$$

Using the additivity property we see that

$$\sum_{k=-\infty}^{\infty} x[k]\,h[n-k] \xrightarrow{\mathcal{T}} \sum_{k=-\infty}^{\infty} x[k]\,h[n-k] .$$

In other words, we have just derived the **input-output relationship** for any LTI system in terms of its **impulse response** as follows:

$$\boxed{x[n] \to \boxed{\text{LTI system with impulse response } h[n]} \to y[n] = \sum_{k=-\infty}^{\infty} x[k]\,h[n-k] .}$$

This sum is so important that it is given its own symbol and name. It is called the **convolution sum** and we write:

$$(x * h)[n] = x[n] * h[n] \triangleq \sum_{k=-\infty}^{\infty} x[k]\,h[n-k] .$$

Expressed compactly:

$$x[n] \to \boxed{\begin{matrix}\text{LTI}\\ h[n]\end{matrix}} \to y[n] = x[n] * h[n] .$$

The fact that all LTI systems have an input-output relationship described by convolution is a tremendous simplification! When designing LTI systems, *i.e.*, filters, we can focus our efforts on designing the impulse response $h[n]$ so that the system behaves according to whatever properties are desired.

Example. Consider a system with given impulse response

$$h[n] = 100\delta[n-7] .$$

What does this system do to a generic input signal $x[n]$? According to the convolution sum:

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]\,h[n-k] = \sum_{k=-\infty}^{\infty} x[k]\,100\delta[n-7-k] = \cdots + 0 + x[n-7]\,100 \cdot 1 + 0 + \cdots = 100x[n-7] ,$$

because $\delta[n-7-k]$ is zero except when its argument is zero, *i.e.*, when $n-7-k=0$ so $k=n-7$.

So what does this system do? It amplifies the signal by 100, and delays it by 7 samples.

What order system is this? 0th order

What about a system with impulse responses $h[n] = \delta[n] + \delta[n-7]$? This is 7th order, just like $1 + x^7$ is a 7th order polynomial.

Sanity check for FIR filters

We had previously stated that for FIR filters the impulse response is

$$h[n] = \sum_{k=M_1}^{M_2} b_k \delta[n-k] \,.$$

What is the corresponding input-output relationship?

$$\begin{aligned} y[n] &= x[n] * h[n] = \sum_{k'=-\infty}^{\infty} x[k'] \, h[n-k'] = \sum_{k'=-\infty}^{\infty} x[k'] \left(\sum_{k=M_1}^{M_2} b_k \delta[n-k'-k] \right) \\ &= \sum_{k=M_1}^{M_2} b_k \left(\sum_{k'=-\infty}^{\infty} x[k'] \, \delta[n-k'-k] \right) = \sum_{k=M_1}^{M_2} b_k x[n-k] \,, \end{aligned}$$

because $\delta[n-k'-k]$ is zero except when its argument is zero, *i.e.*, when $n-k'-k=0$ so $k'=n-k$. This is the input-output relationship presented previously for a generic FIR filter, confirming that convolution of the input signal $x[n]$ with the impulse response $h[n]$ is consistent with our previous formula.

Convolution example

Example. Determine the output signal $y[n]$ when the input signal $x[n]$ is given as follows:

$$x[n] = u[n] + u[n-3] \rightarrow \boxed{\text{LTI with } h[n] = \delta[n] - \delta[n-1]} \rightarrow y[n] = ?$$

In lecture, three solution approaches were discussed to find $y[n] = x[n] * h[n]$.

1. Graphical convolution, using the following three-step recipe
- flip: draw $h[-k]$
- draw $x[k]$ vs k rather than n
- slide: shift $h[-k]$ by n samples for some n
  (shift to the right if n is positive, since $h[n-k] = h[-(k-n)]$)
- Multiply the shifted $h[-k]$ by $x[k]$ sample-by-sample, then sum. This yields $y[n]$ for a particular n.
- Repeat for all integer values of n.

2. Finding the input-output relationship, which is $y[n] = x[n] - x[n-1]$ and examining how this relationship affects the particular input signal $x[n]$ given above graphically.

This system computes the **first difference** of the input signal, which is a discrete-time analog of the operation of taking the derivative of a continuous-time signal. It is useful for **edge detection** in image processing.

3. Substituting in this particular $x[n]$ into the above input-output relationship and simplifying to find that

$$y[n] = \delta[n] + \delta[n-3] \,,$$

using the handy property:

$$\delta[n] = u[n] - u[n-1] \,,$$

which was demonstrated graphically.

All three solution methods are useful.

Properties of convolution and the interconnection of LTI systems

Skill: ***Use properties to simplify LTI systems.*** Awareness of these properties necessary for efficient designs.

Support If $x[n]$ has support $n = N_1, \ldots, N_1 + L_1 - 1$ (length L_1)
and $h[n]$ has support $n = N_2, \ldots, N_2 + L_2 - 1$ (length L_2)
then $y[n] = x[n] * h[n]$ has support $n = N_1 + N_2, \ldots, N_1 + L_1 - 1 + N_2 + L_2 - 1$ (length L_2)
What is the duration of $y[n]$? $L = L_1 + L_2 - 1$

Time-shift

$$x[n] * h[n] = y[n] \quad \Rightarrow \quad \begin{array}{c} x[n-n_0] * h[n] = y[n-n_0] \\ x[n-n_1] * h[n-n_2] = y[n-n_1-n_2] \end{array}$$

Commutative law

$$\boxed{x[n] * h[n] = h[n] * x[n]}$$

Proof:

$$x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]\, h[n-k] = \sum_{k'=-\infty}^{\infty} x[n-k']\, h[k'] = h[n] * x[n]\,,$$

where $k' = n - k$.

Associative law

$$\boxed{(x[n] * h_1[n]) * h_2[n] = x[n] * (h_1[n] * h_2[n])}$$

Proof: let $y_1[n] = (x[n] * h_1[n]) * h_2[n]$ and $y_2[n] = x[n] * (h_1[n] * h_2[n])$.
We must show $y_1[n] = y_2[n]$.

$$\begin{aligned} y_1[n] &= \sum_{k=-\infty}^{\infty} (x * h_1)[k]\, h_2[n-k] = \sum_k \left(\sum_l x[l]\, h_1[k-l] \right) h_2[n-k] = \sum_l x[l] \left(\sum_k h_1[k-l]\, h_2[n-k] \right) \\ &= = \sum_l x[l] \left(\sum_m h_1[m]\, h_2[n-l-m] \right) \sum_l x[l]\ (h_1 * h_2)[n-l] = (x * [h_1 * h_2])[n] = y_2[n]\,, \end{aligned}$$

where $m = k - l$.

The above laws hold in general for any number of systems connected in **series**. So the following notation is acceptable:

$$h[n] = h_1[n] * h_2[n] * \cdots * h_k[n]\,.$$

In particular:

$$\begin{aligned} (x * h_1) * h_2 &= x * (h_1 * h_2) \\ &= x * (h_2 * h_1) \\ &= (x * h_2) * h_1 \end{aligned}$$

so order of serial connection of LTI systems with impulse response h_1 and h_2 does not affect output signal. See picture.

Distributive law

$$\boxed{x[n] * (h_1[n] + h_2[n]) = (x[n] * h_1[n]) + (x[n] * h_2[n])}$$

Proof:

$$\begin{aligned} x[n] * (h_1[n] + h_2[n]) &= \sum_{k=-\infty}^{\infty} x[n-k]\, (h_1[k] + h_2[k]) \\ &= \sum_{k=-\infty}^{\infty} x[n-k]\, h_1[k] + \sum_{k=-\infty}^{\infty} x[n-k]\, h_2[k] \\ &= x[n] * h_1[n] + x[n] * h_2[n]\,. \end{aligned}$$

All of the above follow from simple properties of addition and multiplication due to LTI assumption.

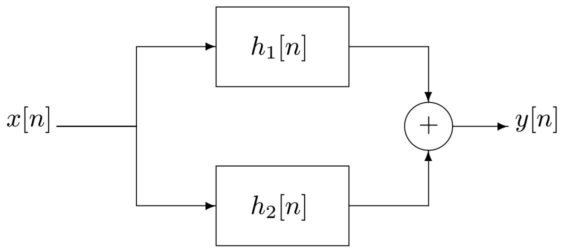
Here are the above properties illustrated with block diagrams.

Commutative
$$x[n] \to \boxed{h[n]} \to y[n]$$
$$h[n] \to \boxed{x[n]} \to y[n]$$
yields same output!

The order of interconnection of systems in **series** or **cascade** does not affect the output[3].

$$x[n] \to \boxed{h_1[n]} \to \boxed{h_2[n]} \to y[n]$$
Associative
$$x[n] \to \boxed{h_1[n] * h_2[n]} \to y[n]$$
Commutative
$$x[n] \to \boxed{h_2[n] * h_1[n]} \to y[n]$$
Associative
$$x[n] \to \boxed{h_2[n]} \to \boxed{h_1[n]} \to y[n]$$

Parallel connection:



Distributive: $x[n] \to \boxed{h[n] = h_1[n] + h_2[n]} \to y[n]$

Example.

$$x[n] \Longrightarrow \boxed{h_1[n] = \delta[n] - \delta[n-1]} \to \boxed{h_2[n] = u[n]} \to y[n]$$

Overall impulse response:

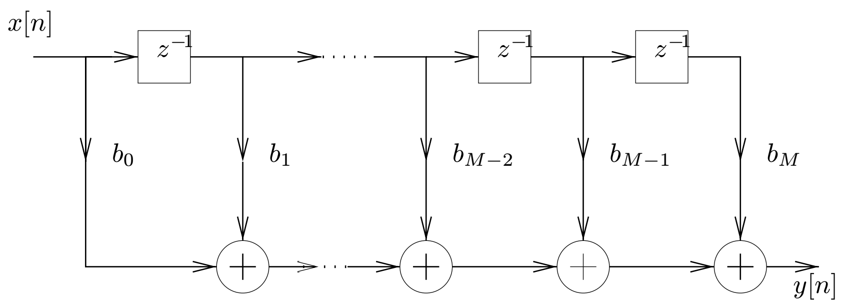
$$h[n] = h_1[n] * h_2[n] = (\delta[n] - \delta[n-1]) * u[n] = u[n] - u[n-1] = \delta[n]\,.$$

Block diagram of FIR filters

An Mth order filter has $M-1$ unit delays (why z^{-1} explained later), $M+1$ multiplies, and M adds.

This could be implemented in
- digital hardware (adders, multipliers)
  How are delays implemented? With buffers / registers / latches / flip-flops.
- software on general purpose computer (*e.g.*, ANSI C, MATLAB, etc.)

[3]This claim only holds for ideal LTI systems based on real numbers. In digital systems where the signal values and filter coefficients are quantized, the order of interconnection can matter in some cases.


$x[n]$
z^{-1}
z^{-1}
z^{-1}
b_0
b_1
b_{M-2}
b_{M-1}
b_M
$y[n]$

Convolution with impulses

Time shift / delay:

$$x[n] * \delta[n-n_0] = x[n-n_0]$$

Identity:

$$x[n] * \delta[n] = x[n]$$

Cascade of time shifts:

$$\delta[n-n_1] * \delta[n-n_2] = \delta[n-n_1-n_2]$$

Properties of LTI systems in terms of the impulse response

Since an LTI system is completely characterized by its impulse response, we should be able to express the properties of causality and stability in terms of $h[n]$.

Causal LTI systems

Recall system is causal iff output $y[n]$ depends only on present and past values of input.

For an LTI system with impulse response $h[n]$:

$$y[n] = \sum_{k=-\infty}^{\infty} h[k]\, x[n-k] = \sum_{k=0}^{\infty} h[k]\, x[n-k] + \sum_{k=-\infty}^{-1} h[k]\, x[n-k] \,.$$

The first sum depends on present and past input samples $x[n]\,, x[n-1]\,, \ldots$, whereas the second sum depends on future input samples $x[n+1]\,, x[n+1]\,, \ldots$.
Thus the system is causal iff the impulse response terms corresponding to the second sum is zero.
These terms are $h[-1]\,, h[-2]\,, \ldots$.

$$\boxed{\text{An LTI system is } \textbf{causal} \text{ iff its impulse response } h[n] = 0 \text{ for all } n < 0.}$$

In the causal case the convolution summation simplifies slightly since we can drop the right sum above:

$$y[n] = \sum_{k=0}^{\infty} h[k]\, x[n-k] = \sum_{k=-\infty}^{n} x[k]\, h[n-k] \,.$$

Example. Is the LTI system with $h[n] = u(n-n_0-5)$ causal? Only if $n_0 + 5 \geq 0$.

A **causal sequence** is a sequence $x[n]$ which is zero for all $n < 0$.

If the input to a **causal** LTI system is a **causal sequence**, then the output is simply

$$y[n] = \begin{cases} 0, & n < 0 \\ \sum_{k=0}^{n} h[k]\, x[n-k] = \sum_{k=0}^{n} x[k]\, h[n-k]\,, & n \geq 0. \end{cases}$$

The above sum is precisely what is computed by MATLAB's `conv` function, for finite-length $x[n]$ and $h[n]$.

Stability of LTI systems

Recall $y[n] = \sum_{k=-\infty}^{\infty} h[k]\, x[n-k]$ so by the triangle inequality

$$|y[n]\,| = \left|\sum_{k=-\infty}^{\infty} h[k]\, x[n-k]\right| \leq \sum_{k=-\infty}^{\infty} |h[k]\, x[n-k]| = \sum_{k=-\infty}^{\infty} |h[k]\,||x[n-k]\,| \leq M_x \sum_{k=-\infty}^{\infty} |h[k]\,|$$

if $|x[n]\,| \leq M_x \; \forall n$.

Thus, for an LTI system to be BIBO stable, it is sufficient that its impulse response be **absolutely summable**, *i.e.*

$$\sum_{n=-\infty}^{\infty} |h[n]\,| < \infty.$$

It is also *necessary* for the above summation to be finite. To show necessity of that condition, construct counter-example showing absence of the condition means we can construct some bounded input signal for which the output is unbounded. A suitable choice is

$$x[n] = \begin{cases} h^*[-n]\,/|h[-n]\,|, & h[-n] \neq 0 \\ 0, & h[-n] = 0 \end{cases}$$

which is bounded by $M_x = 1$. But

$$y[0] = \sum_{k=-\infty}^{\infty} h[k]\, x[-k] = \sum_{k=-\infty}^{\infty} |h[k]\,| = \infty$$

if the impulse response is not absolutely summable, so the output would not be bounded for the specified bounded input signal.

Thus we have shown

$$\boxed{\text{A LTI system is BIBO stable iff its impulse response is absolutely summable, } \textit{i.e.}, \sum_{n=-\infty}^{\infty} |h[n]| < \infty.}$$

Example: Accumulator: $y[n] = y[n-1] + x[n]$.
What is impulse response? Let $x[n] = \delta[n]$, then $y[n] = u[n]$. So $h[n] = u[n]$.
Alternatively, recall that $\delta[n] = u[n] - u[n-1]$, so $u[n] = u[n-1] + \delta[n]$.
By correspondence with the input-output relationship above, it must be the case that $h[n] = u[n]$.
Stable? No: $\sum_{n=-\infty}^{\infty} h[n] = \infty$, so unstable.

One can also show the following for a BIBO stable system (see text).
- The impulse response $h[n]$ goes to zero as $n \to \infty$.
- If the input $x[n]$ has finite duration, then the output $y[n]$ will decay to zero as $n \to \infty$.

Duration of impulse response

Two classes
- **finite impulse response** or **FIR**
  only a finite number of $h[n]$ are nonzero
- **infinite impulse response** or **IIR**
  an infinite number of $h[n]$ are nonzero