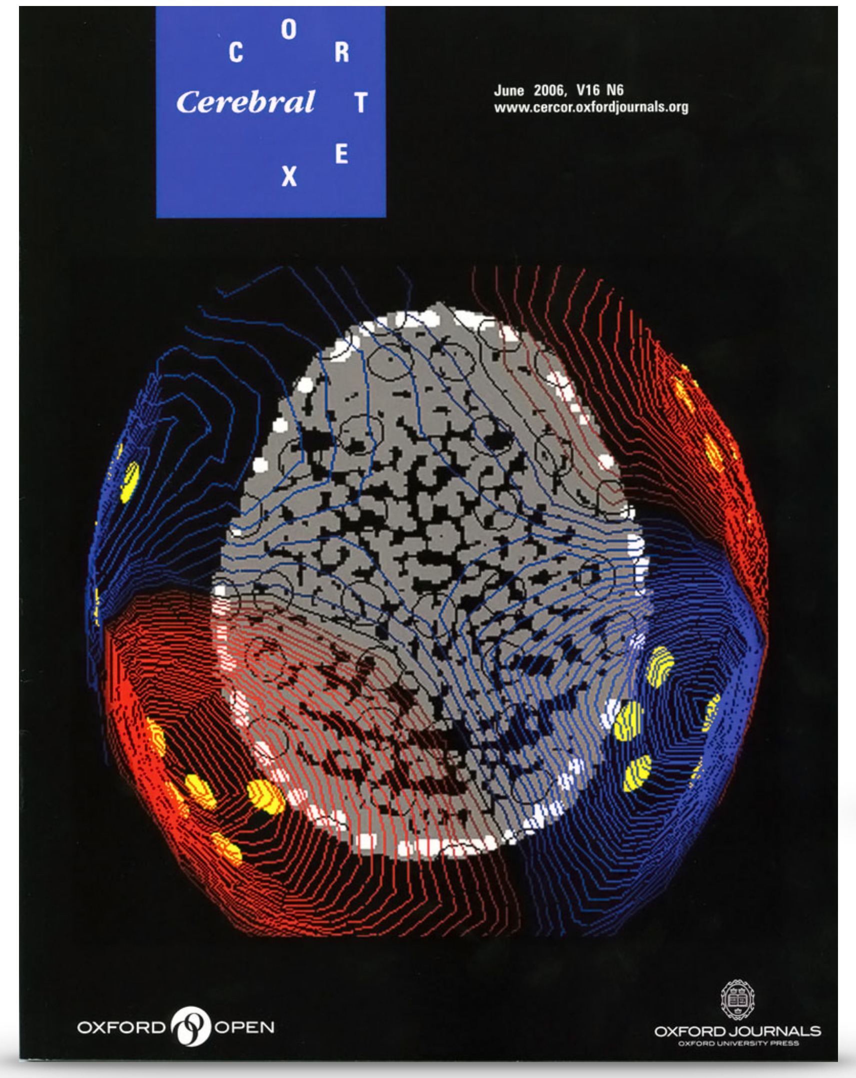
Signal Analysis Primer and Applications

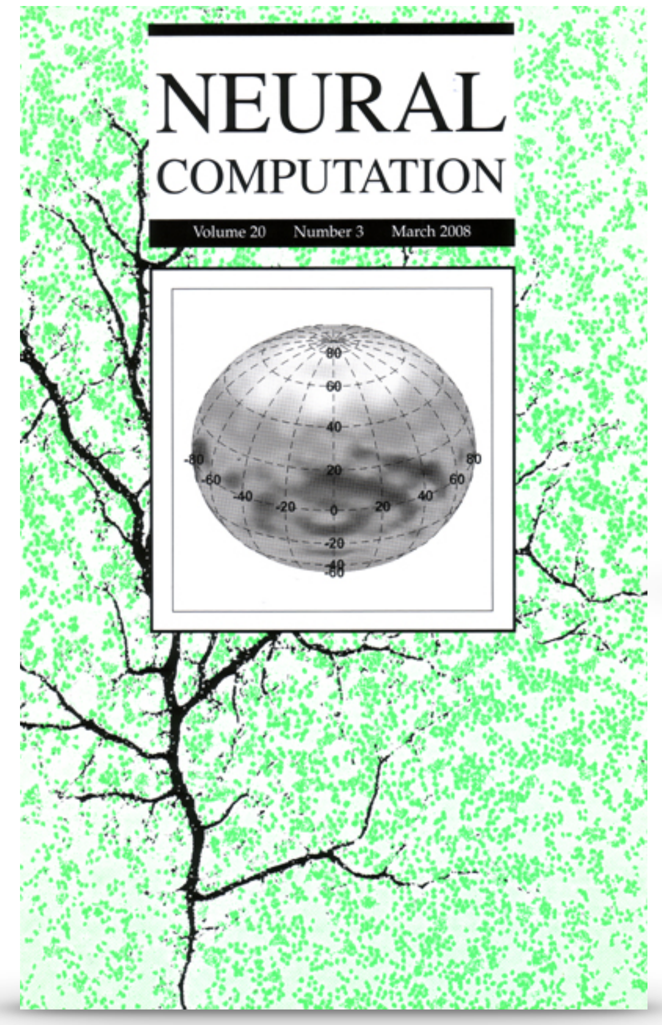
Jonathan Z. Simon
University of Maryland, College Park

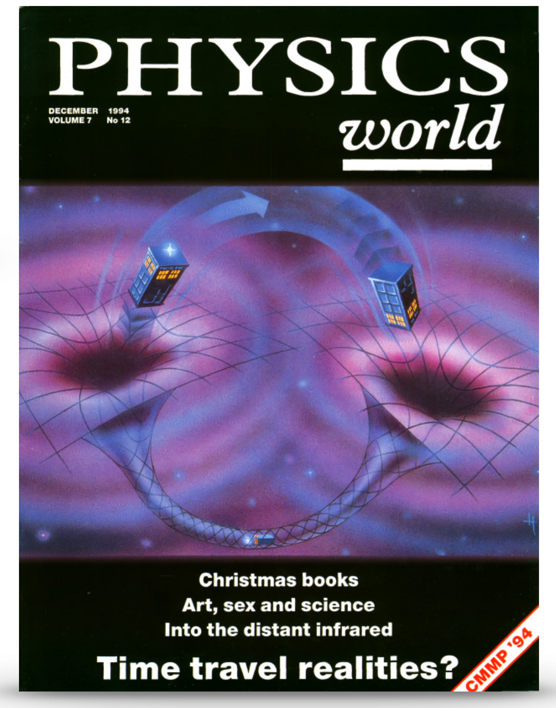
Digital Signal Processing in Neurophysiology
University of Lübeck
10-11 June 2016

Research Background

- MEG-based Auditory Neuroscience
  - Cocktail-Party Auditory Processing
  - Auditory Attention
  - Neural Representations of Speech
  - Fundamentally Temporally Neural Representations
- More at <http://www.isr.umd.edu/Labs/CSSL/simonlab/>







Teaching Background

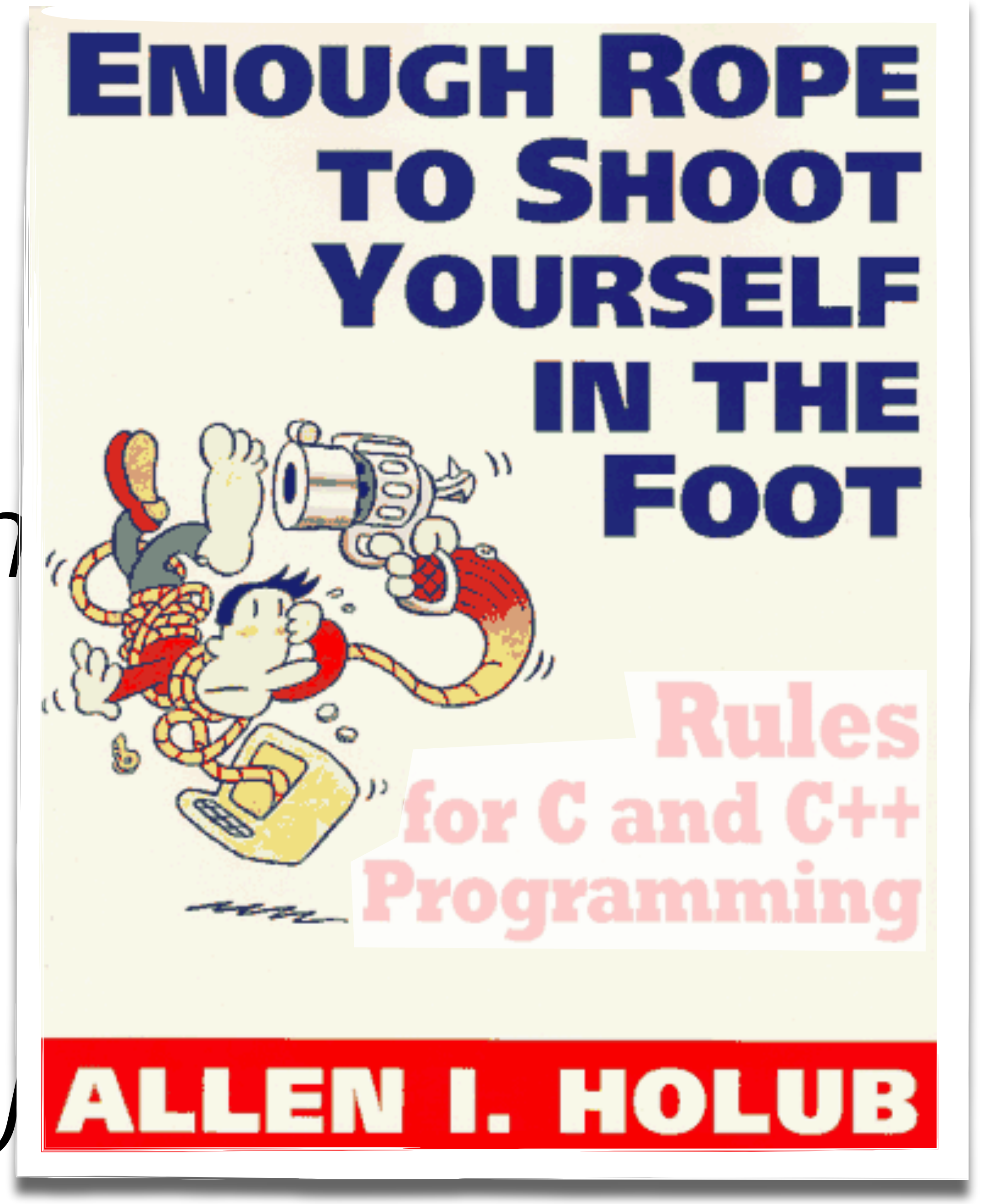
- Courses in Two Departments (with very different students)
    - Electrical & Computer Engineering
    - Biology
- Developed course: “Quantitative Analysis of Biological Data”
  for Neuroscience/Cognitive Neuroscience/Biology graduate students
- Feel **very** free to ask “stupid” questions (they’re **not**).

Outline

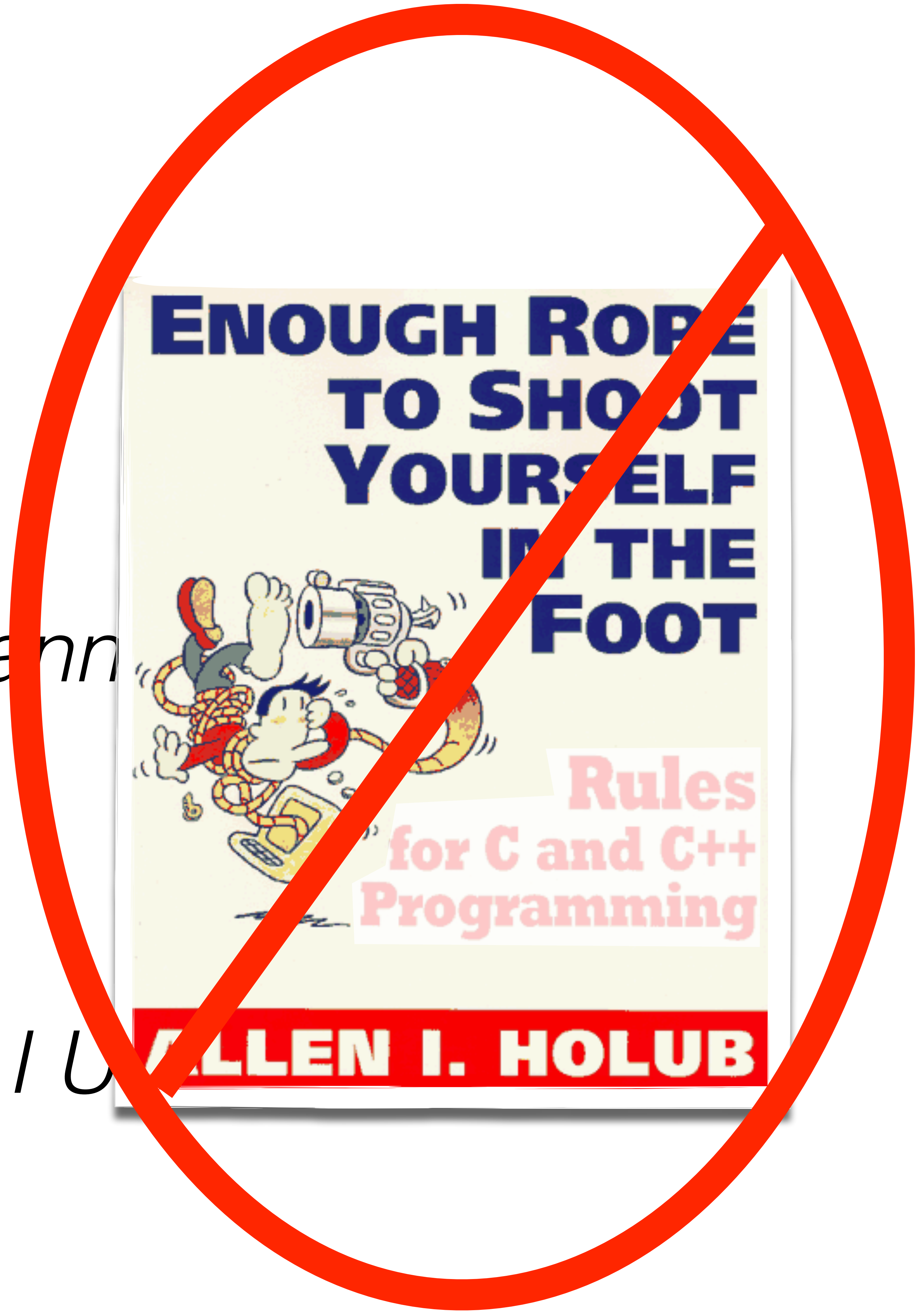
- Fourier Transform: *Why It's Useful*, and *What it Can/Cannot Do For You*
- Filters: *What They Do*, and *How They Do It*
- Filters: *Why So Many Different Kinds? Which Should I Use and When*?
- Grab Bag:
    - *Use Causal Filters; Windowing is Good*

Outline

- Fourier Transform: *Why It's Useful*, and *What it Can/Cann*
- Filters: *What They Do*, and *How They Do It*
- Filters: *Why So Many Different Kinds? Which Should I U*
- Grab Bag:
  - *Use Causal Filters; Windowing is Good*



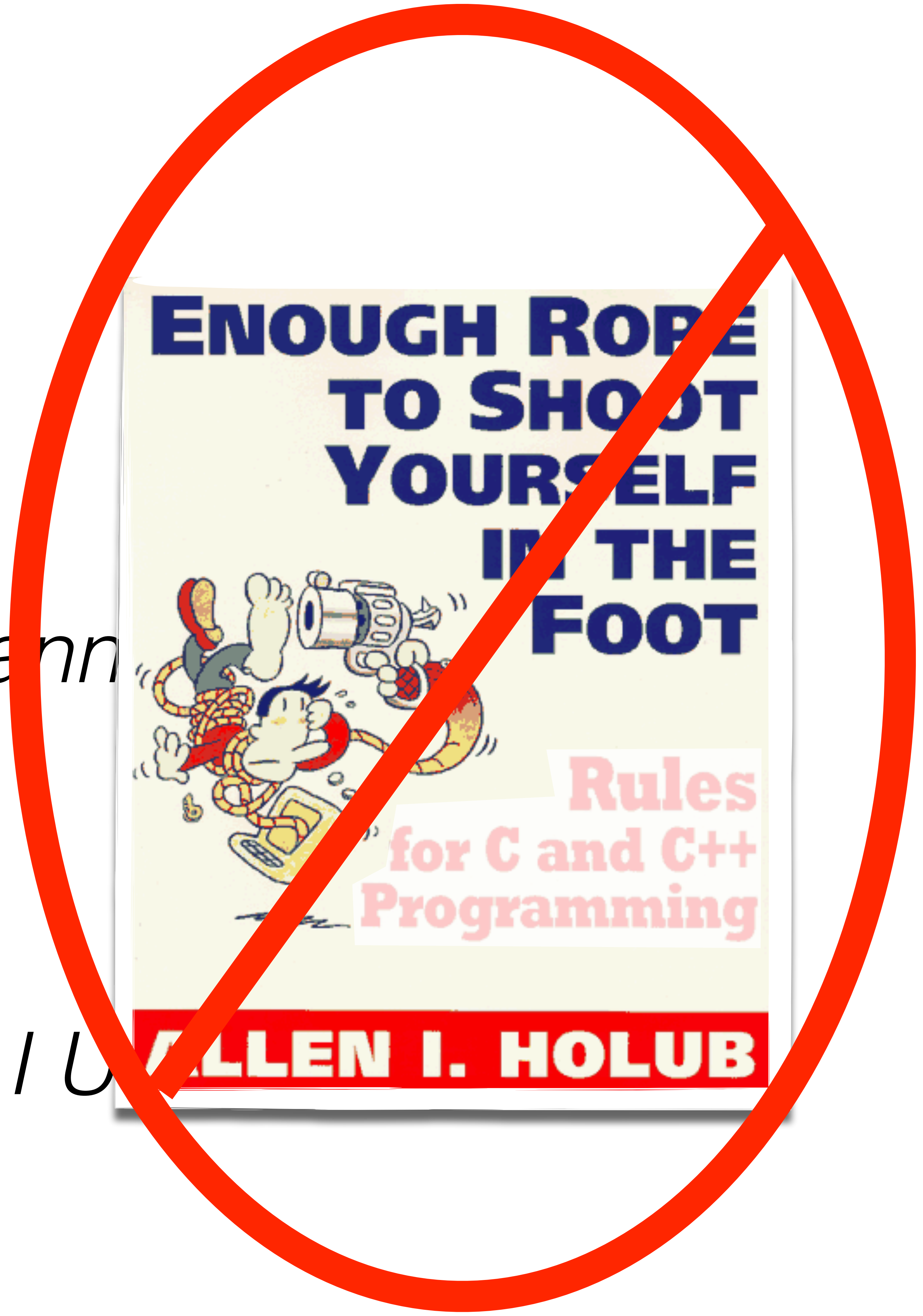
Outline



- Fourier Transform: *Why It's Useful*, and *What it Can/Cann*
- Filters: *What They Do*, and *How They Do It*
- Filters: *Why So Many Different Kinds? Which Should I U*
- Grab Bag:
  - *Use Causal Filters; Windowing is Good*

Outline

- Fourier Transform: *Why It's Useful*, and *What it Can/Cann*
- Filters: *What They Do*, and *How They Do It*
- Filters: *Why So Many Different Kinds?*   *Which Should I U*
- Grab Bag:
    - *Use Causal Filters; Windowing is Good*



Multiple Breaks for Computer Lab Exercises

Outline

- Fourier Transform: *Why It's Useful*, and *What it Can/Cannot Do For You*
- Filters: *What They Do*, and *How They Do It*
- Filters: *Why So Many Different Kinds? Which Should I Use and When*?
- Grab Bag:
  - *Use Causal Filters; Windowing is Good*

The Fourier Transform

- **Every** Time-Domain Signal can be Re-expressed as a Sum of Sinusoids/ Oscillations
- # of time points = # of frequencies
- Reciprocal relationship: *time* resolution (Δt) & *sample frequency* (f_s)
- Reciprocal relationship: *frequency resolution* (Δf) & *duration* (T)

$$x[t] = \frac{1}{N}\sum_{k=0}^{N-1} X[f_k]\mathrm{e}^{i2\pi f_k t} \quad \text{where:}$$

$$t = \underbrace{0, \Delta t, 2\Delta t, \ \ldots \ , T - \Delta t}_{N}$$

$$f_k = \underbrace{0, \Delta f, 2\Delta f, \ \ldots \ , f_s - \Delta f}_{N}$$

$$f_s = \mathit{sampling\ frequency} = \frac{1}{\Delta t}$$

$$T = \mathit{signal\ duration} = \frac{1}{\Delta f}$$

The Fourier Transform

- **Every** Time-Domain Signal can be Re-expressed as a Sum of Sinusoids/ Oscillations
- # of time points = # of frequencies
- Reciprocal relationship: *time resolution* (Δt) & *sample frequency* (f_s)
- Reciprocal relationship: *frequency resolution* (Δf) & *duration* (T)

$$x[t] = \frac{1}{N}\sum_{k=0}^{N-1} X[f_k]\mathrm{e}^{i2\pi f_k t} \quad \text{where:}$$

$$t = \underbrace{0, \Delta t, 2\Delta t, \ \ldots \ , T - \Delta t}_{N}$$

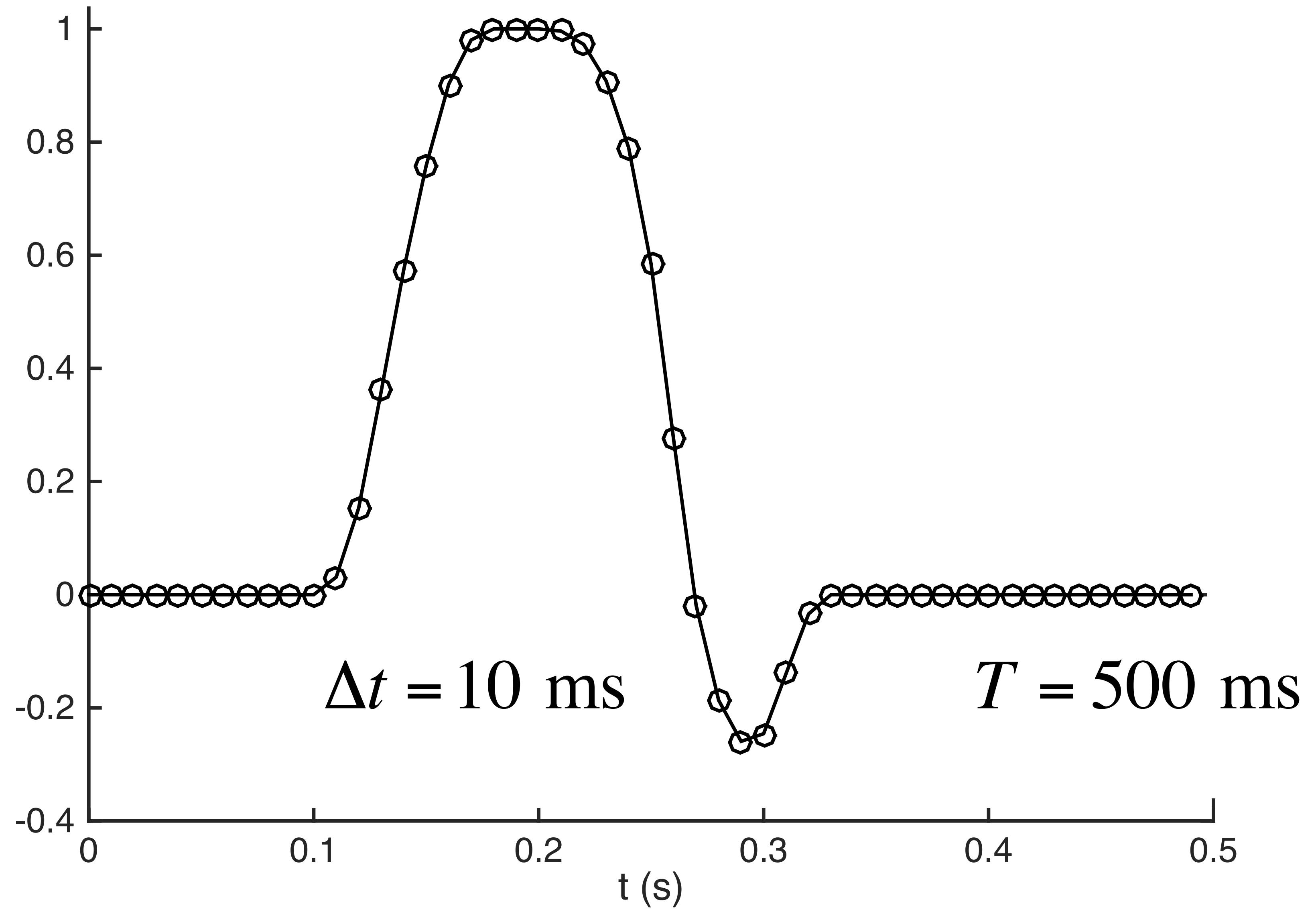
$$f_k = \underbrace{0, \Delta f, 2\Delta f, \ \ldots \ , f_s - \Delta f}_{N}$$

$$f_s = sampling\ frequency = \frac{1}{\Delta t}$$

$$T = signal\ duration = \frac{1}{\Delta f}$$

The Fourier Transform

- **Every** Time-Domain Signal can be Re-expressed as a Sum of Sinusoids/ Oscillations
- # of time points = # of frequencies
- Reciprocal relationship: *time* resolution (Δt) & *sample frequency* (f_s)
- Reciprocal relationship: *frequency resolution* (Δf) & *duration* (T)



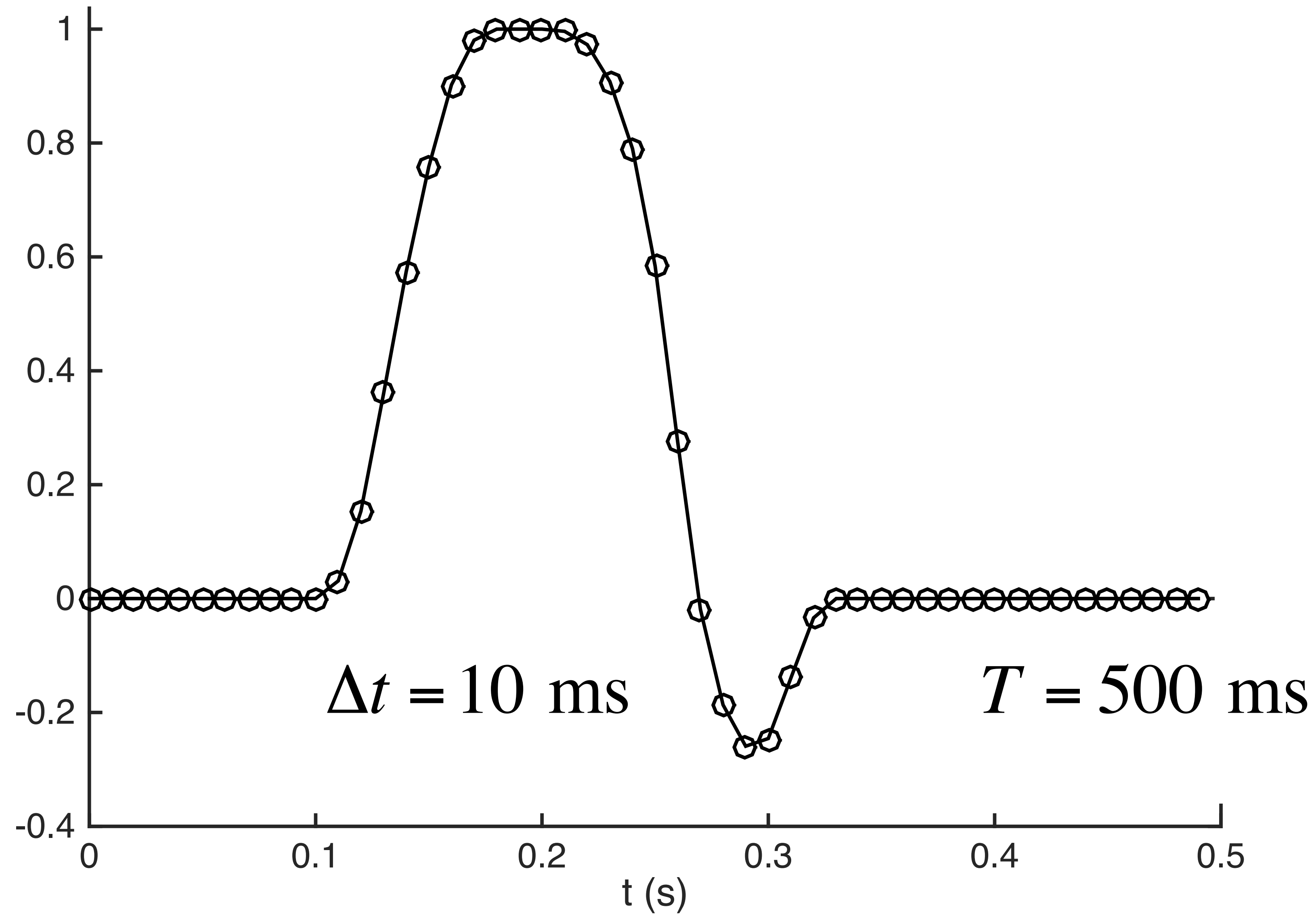
The Fourier Transform

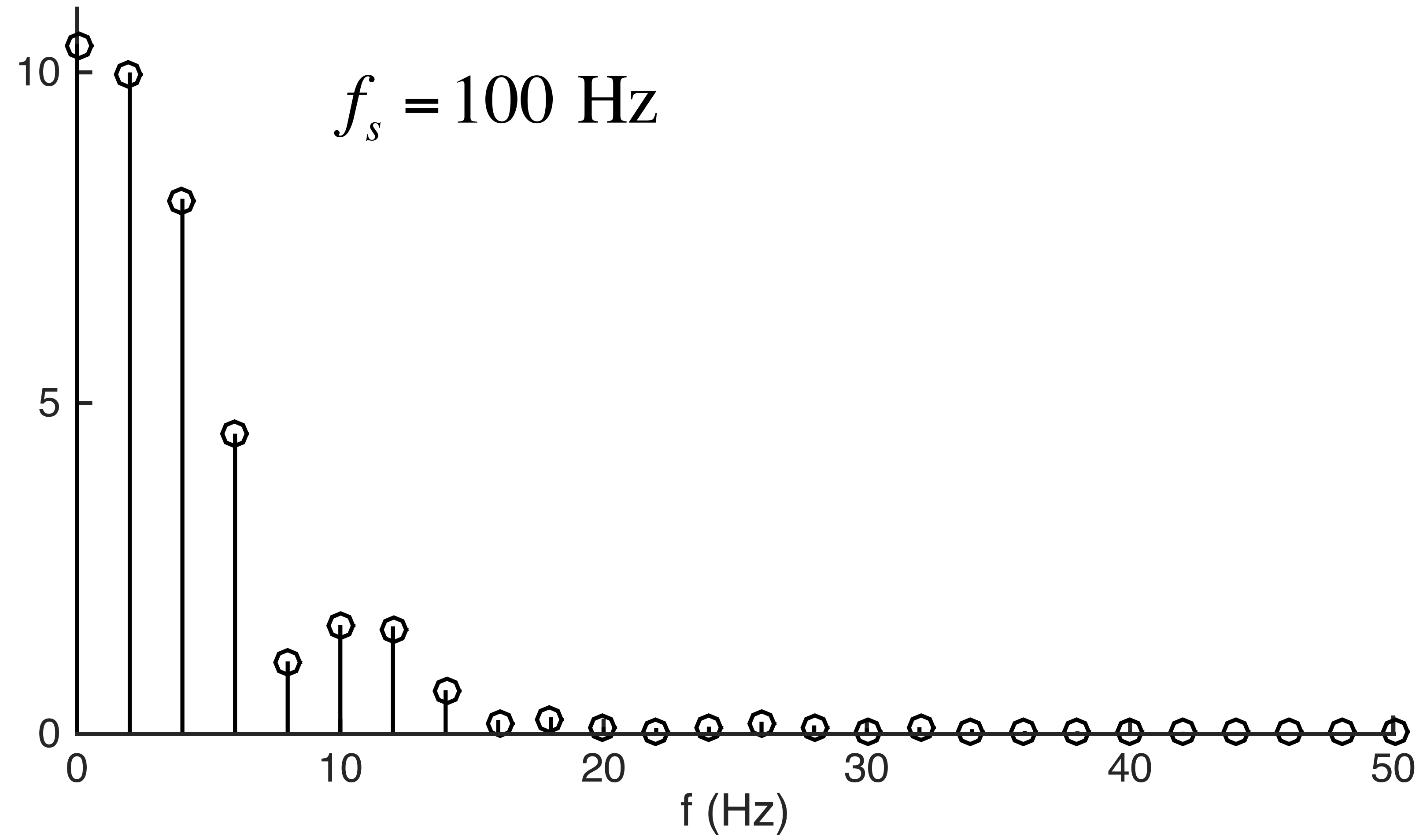
- **Every** Time-Domain Signal can be Re-expressed as a Sum of Sinusoids/ Oscillations

- # of time points = # of frequencies

- Reciprocal relationship: *time* resolution (Δt) & *sample frequency* (f_s)

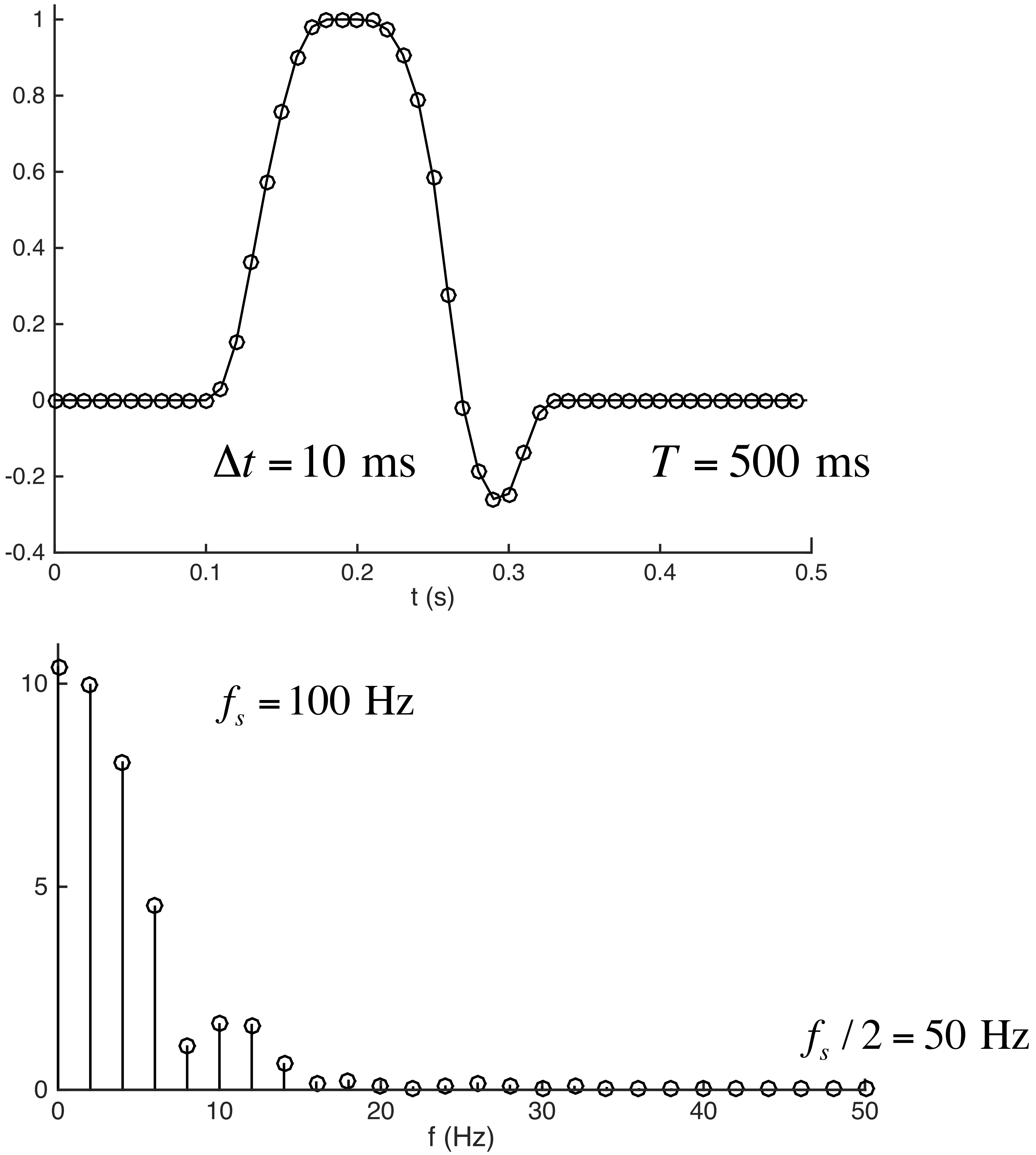
- Reciprocal relationship: *frequency resolution* (Δf) & *duration* (T)





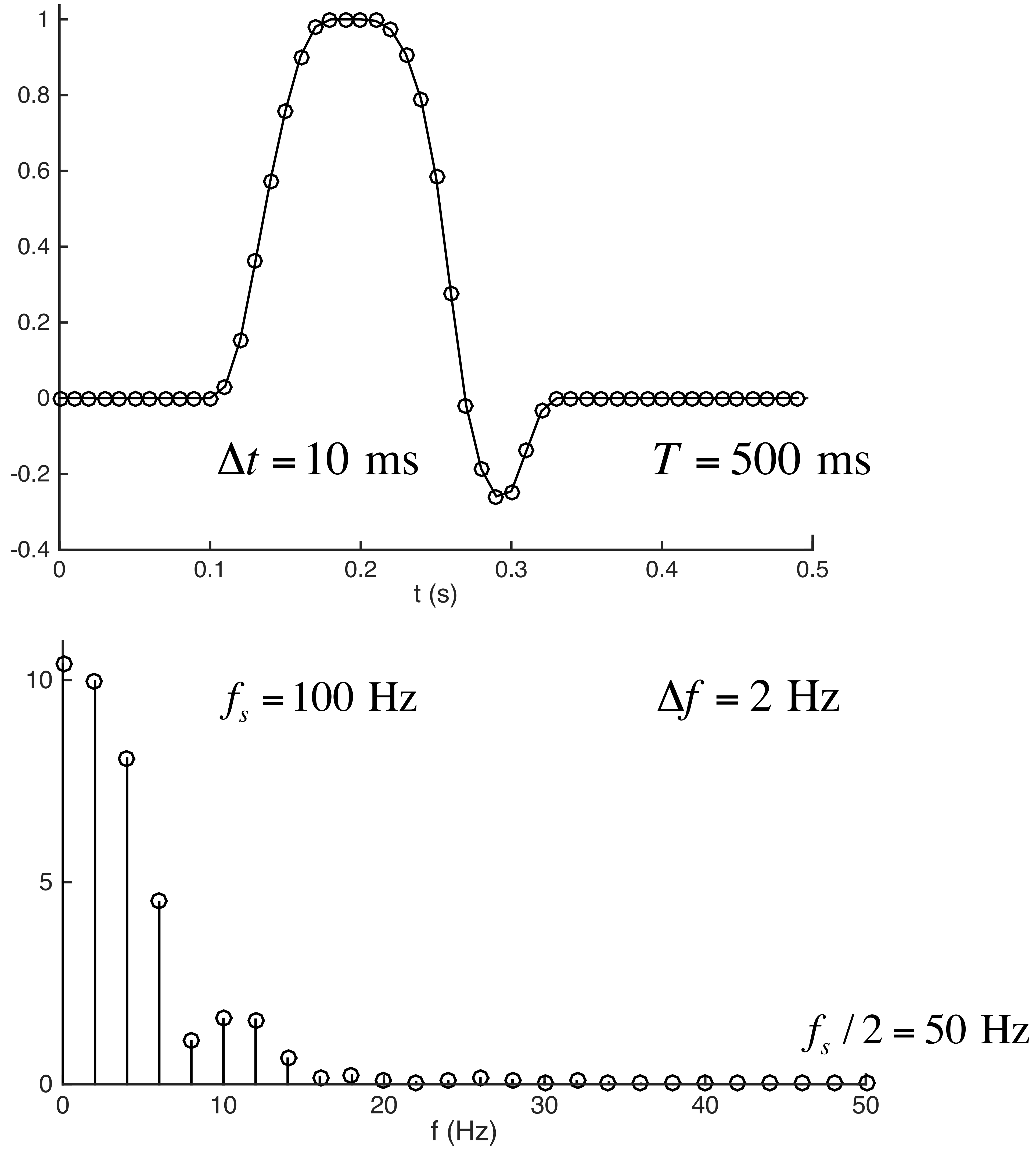
The Fourier Transform

- **Every** Time-Domain Signal can be Re-expressed as a Sum of Sinusoids/ Oscillations
- # of time points = # of frequencies
- Reciprocal relationship: *time resolution* (Δt) & *sample frequency* (f_s)
- Reciprocal relationship: *frequency resolution* (Δf) & *duration* (T)



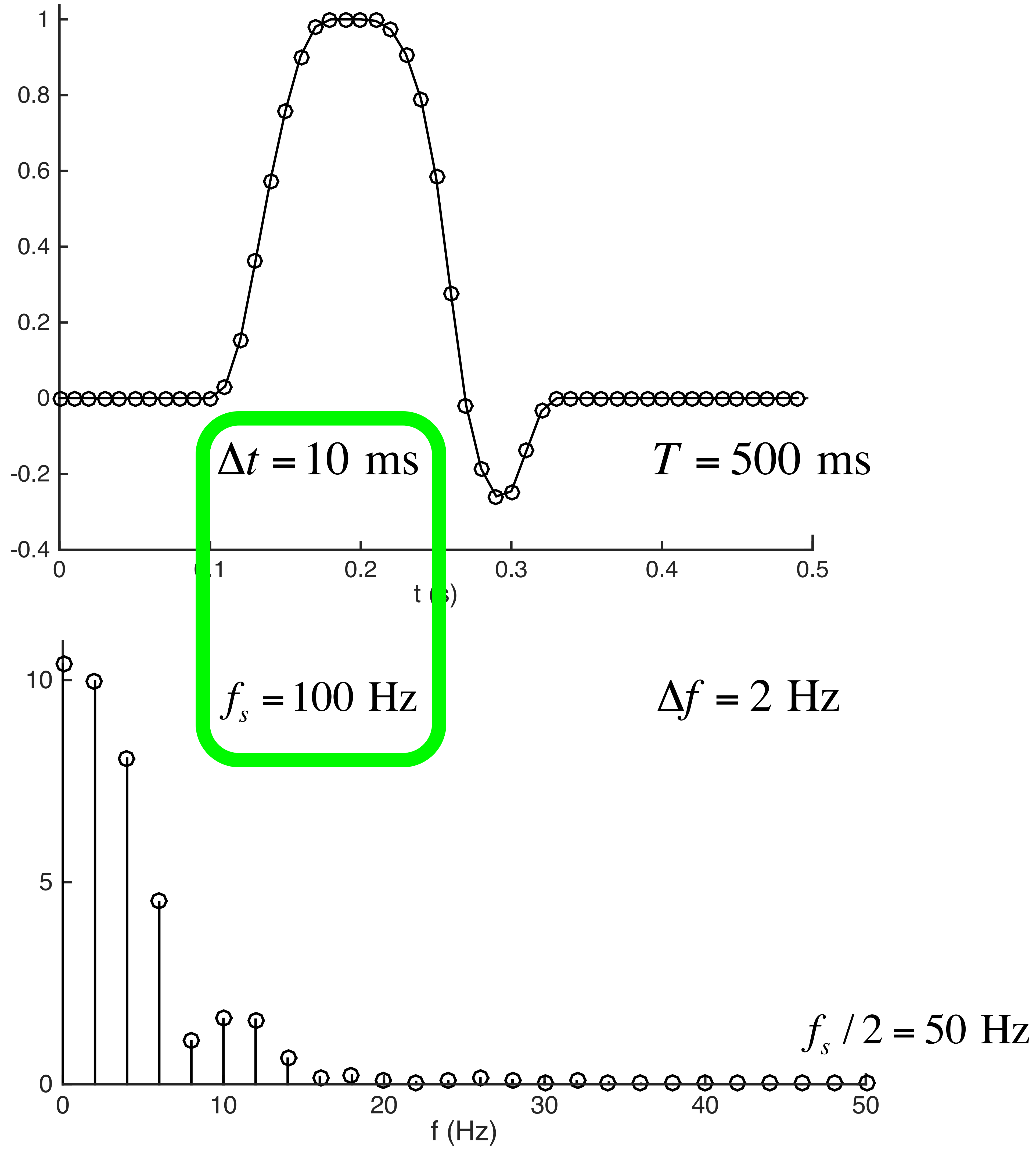
The Fourier Transform

- **Every** Time-Domain Signal can be Re-expressed as a Sum of Sinusoids/ Oscillations
- # of time points = # of frequencies
- Reciprocal relationship: *time* resolution (Δt) & *sample frequency* (f_s)
- Reciprocal relationship: *frequency resolution* (Δf) & *duration* (T)



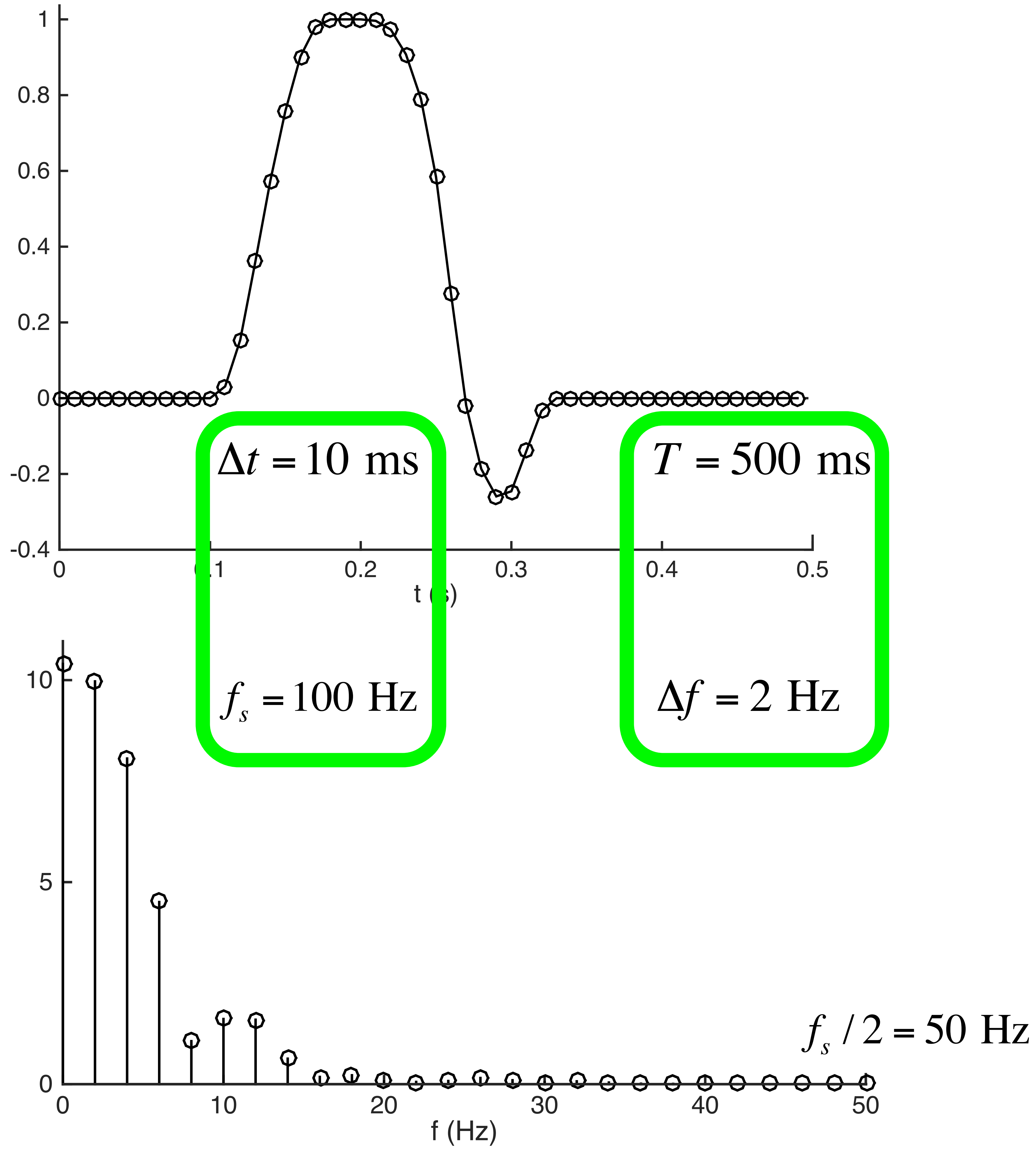
The Fourier Transform

- **Every** Time-Domain Signal can be Re-expressed as a Sum of Sinusoids/ Oscillations
- # of time points = # of frequencies
- Reciprocal relationship: *time resolution* (Δt) & *sample frequency* (f_s)
- Reciprocal relationship: *frequency resolution* (Δf) & *duration* (T)



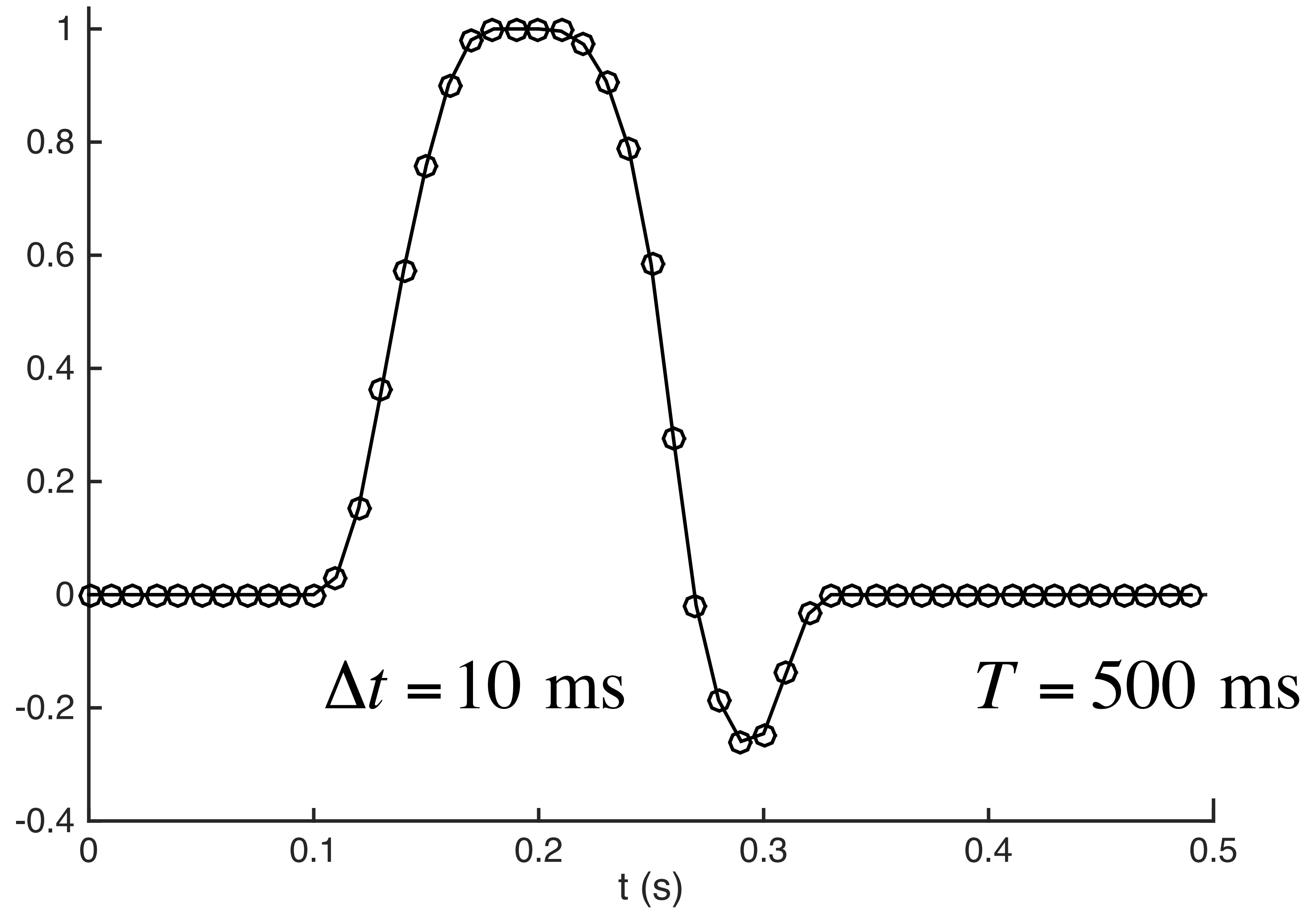
The Fourier Transform

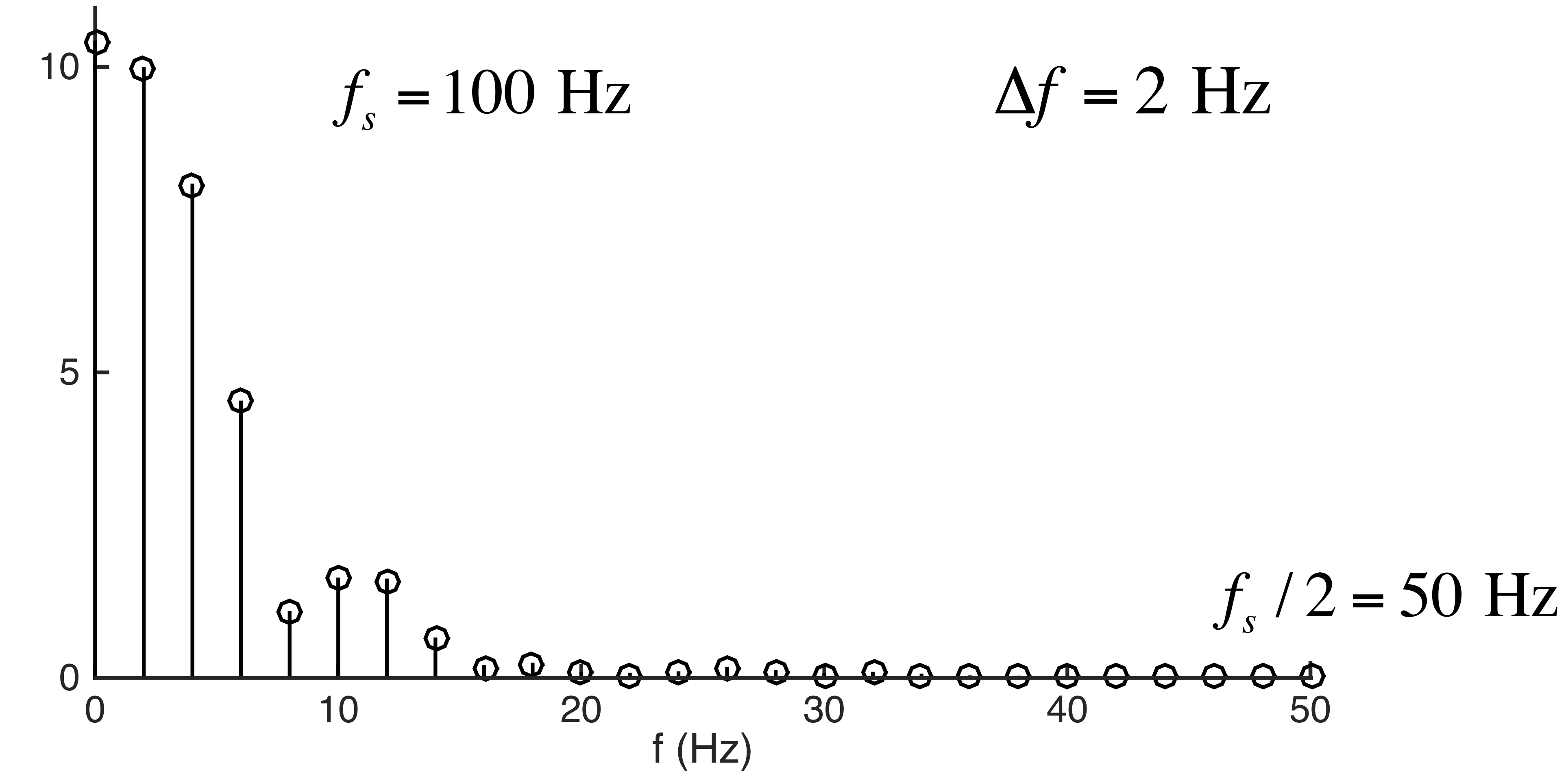
- **Every** Time-Domain Signal can be Re-expressed as a Sum of Sinusoids/ Oscillations
- # of time points = # of frequencies
- Reciprocal relationship: *time resolution* (Δt) & *sample frequency* (f_s)
- Reciprocal relationship: *frequency resolution* (Δf) & *duration* (T)



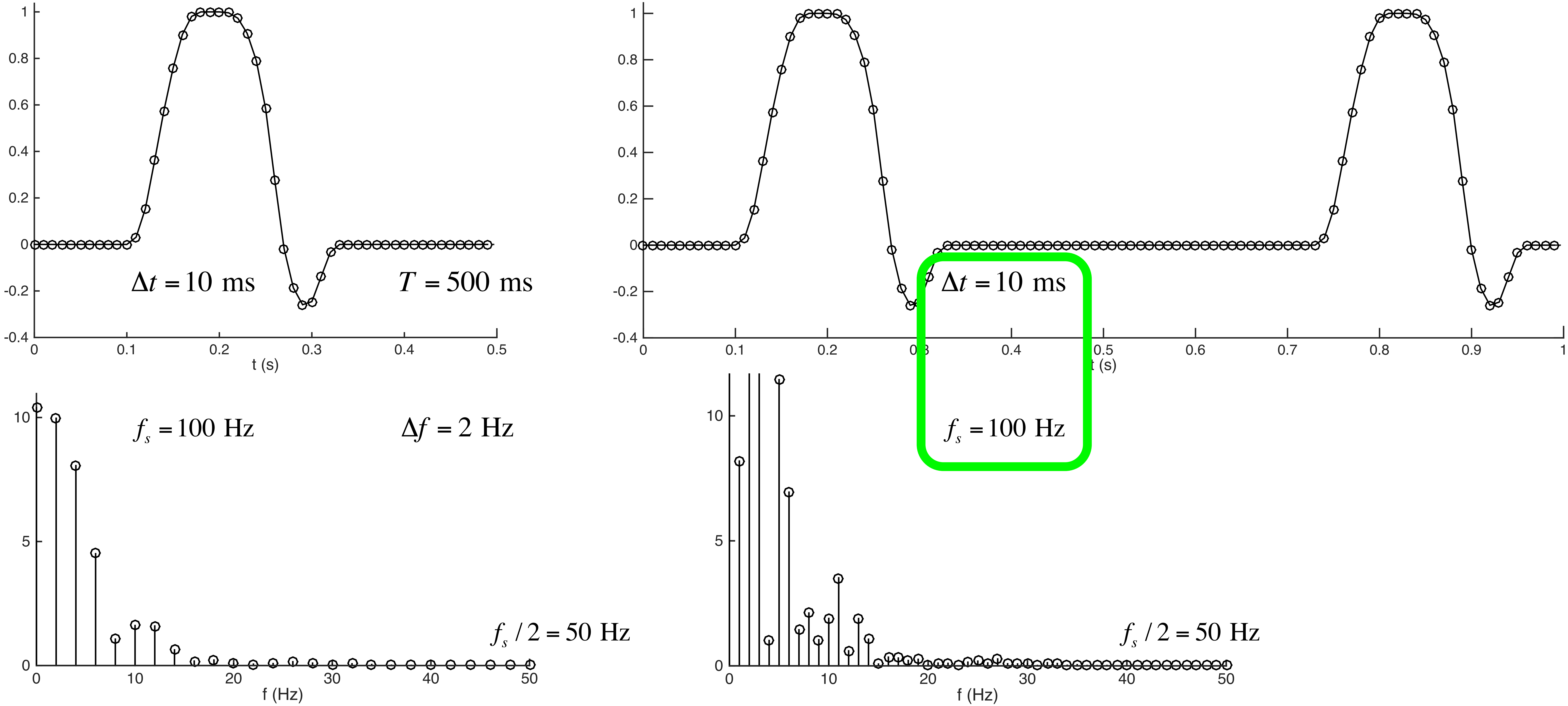
Fourier Transform: Time-Frequency Tradeoff

- **Every** Time-Domain Signal can be Re-expressed as a Sum of Sinusoids/ Oscillations
- # of time points = # of frequencies
- Reciprocal relationship: *time* resolution (Δt) & *sample frequency* (f_s)
- Reciprocal relationship: *frequency resolution* (Δf) & *duration* (T)

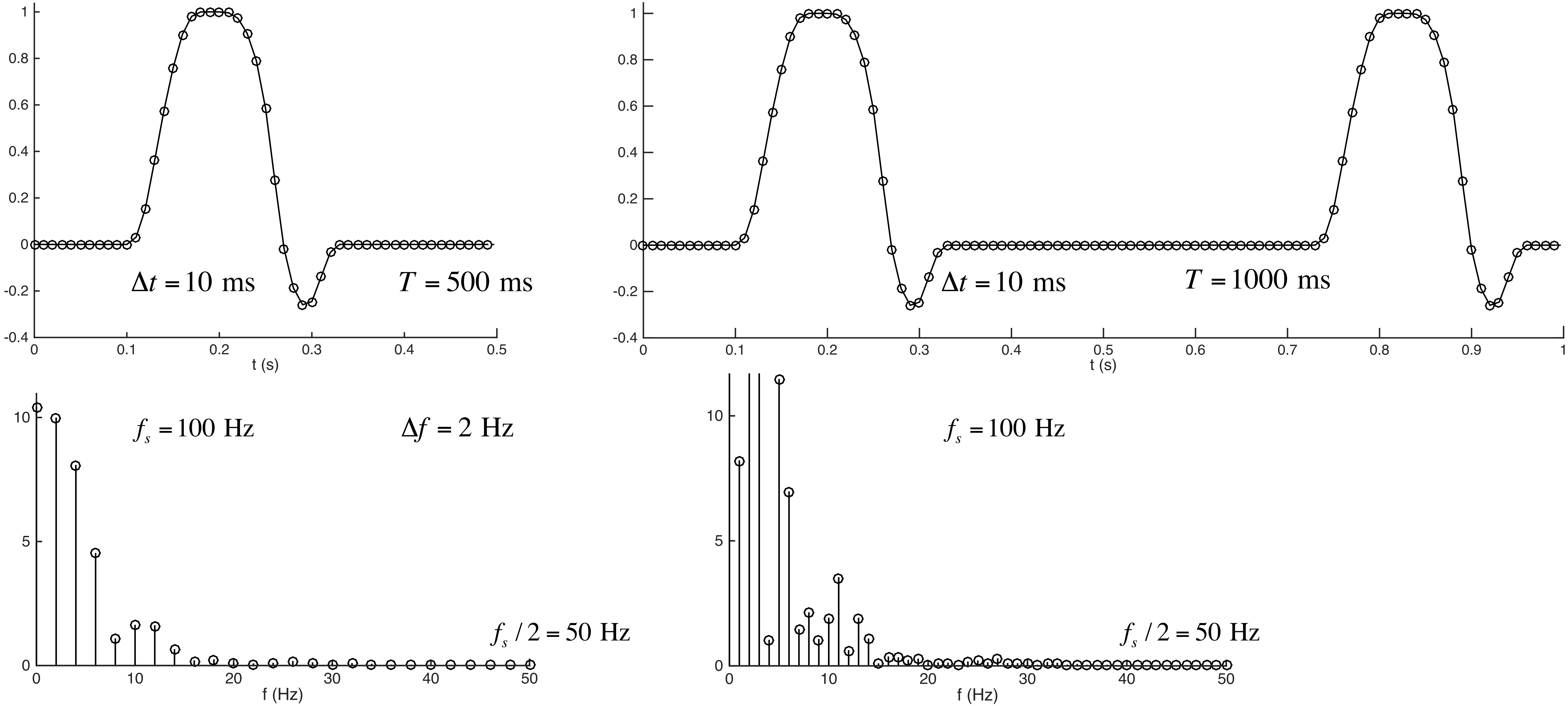




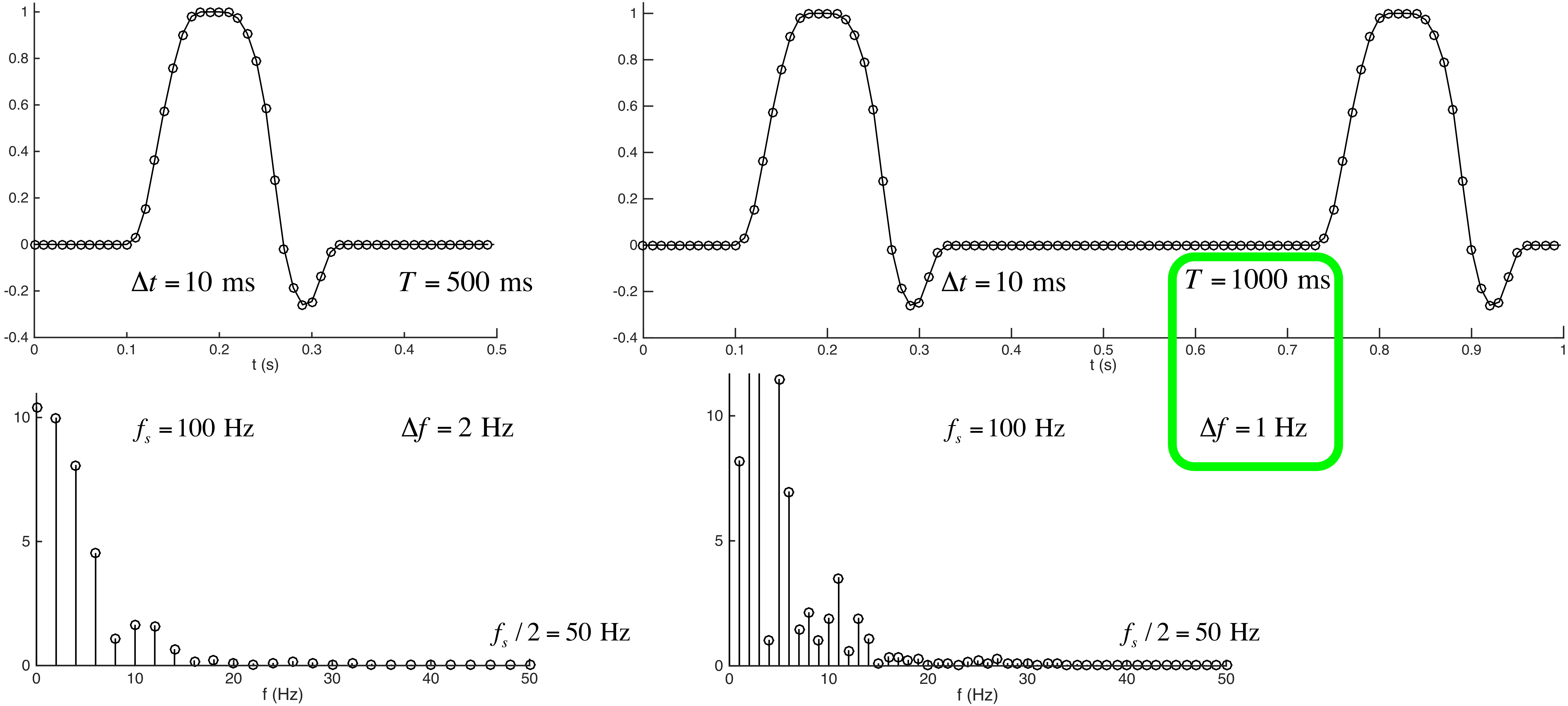
Fourier Transform: Time-Frequency Tradeoff



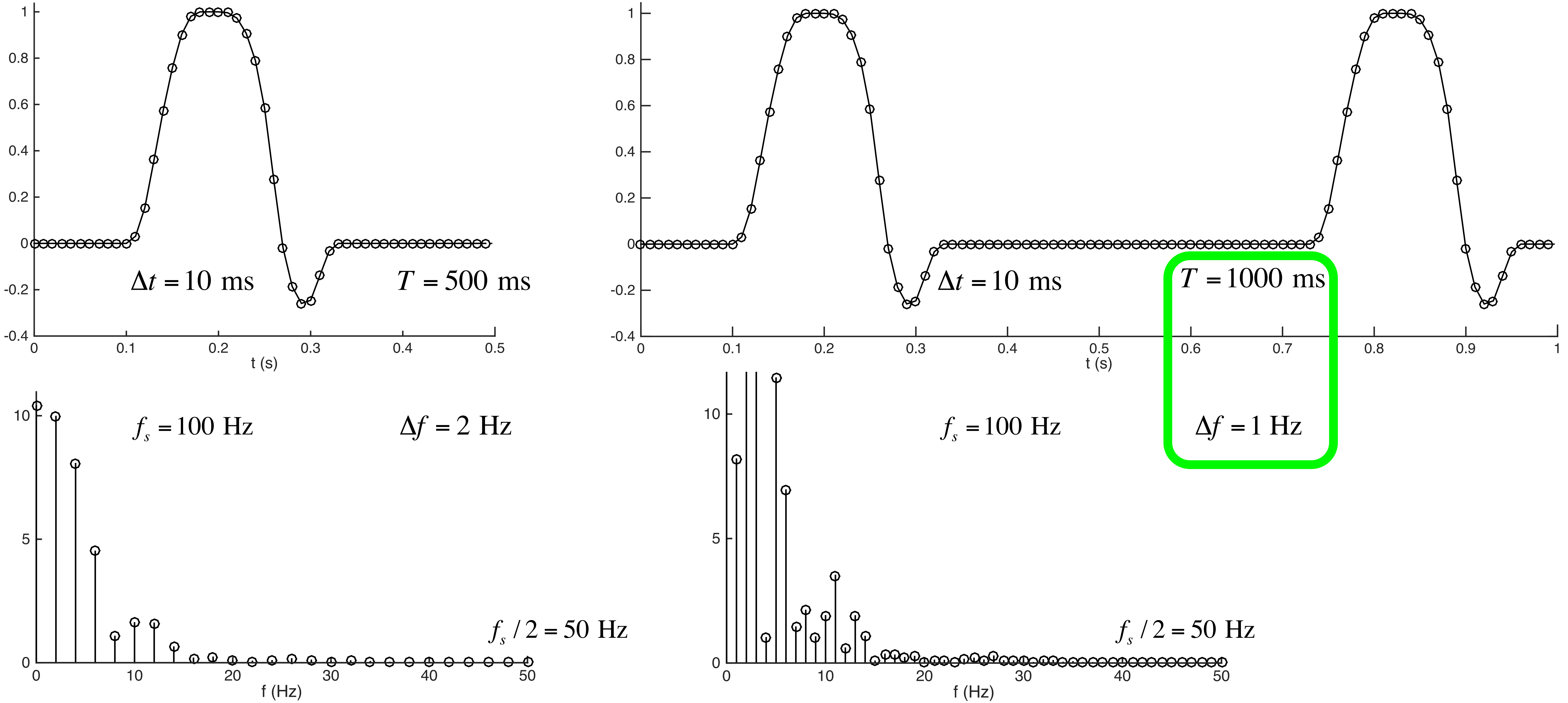
Fourier Transform: Time-Frequency Tradeoff



Fourier Transform: Time-Frequency Tradeoff

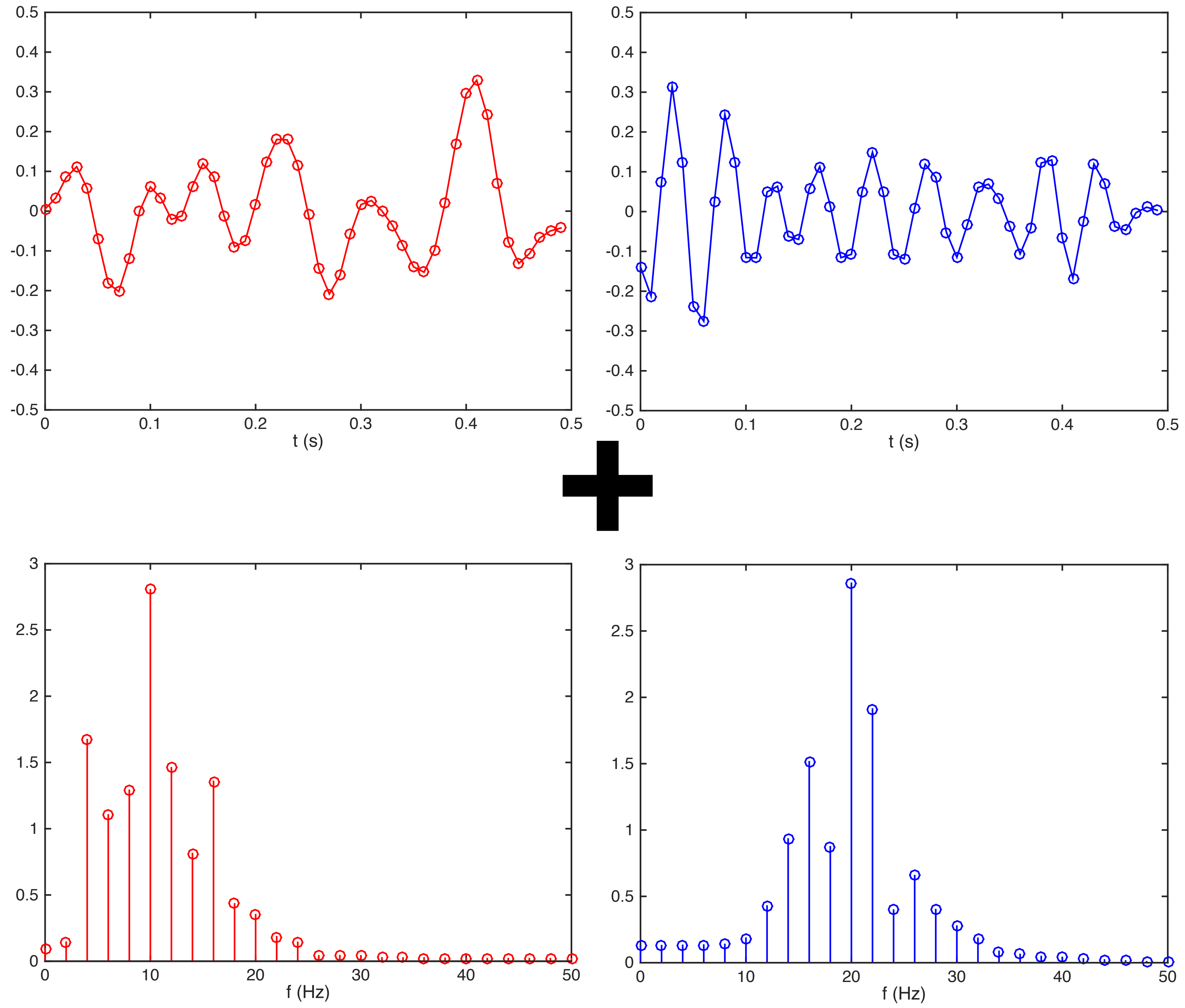


Break for Computer Lab Exercise 1



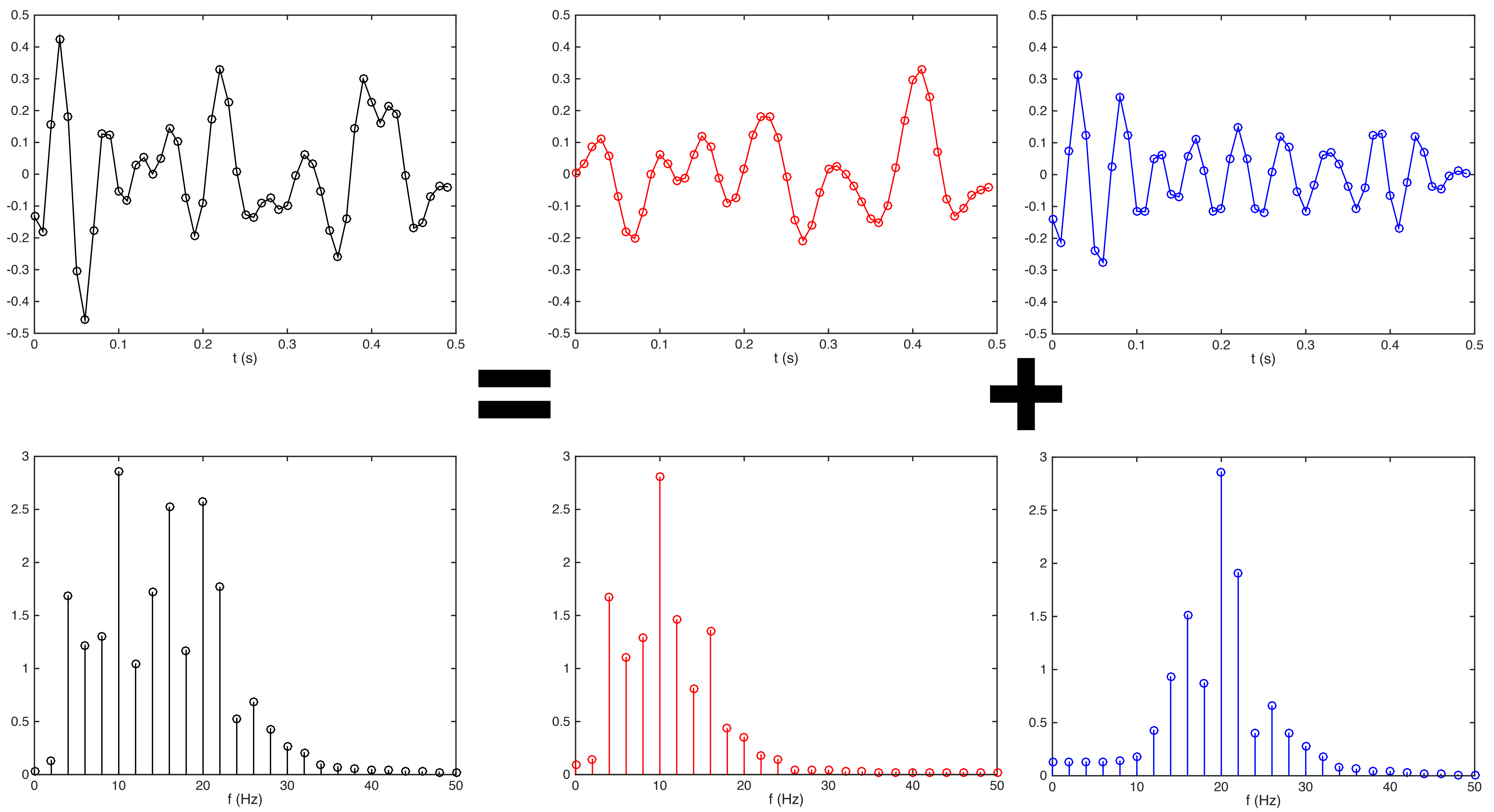
Fourier Transform: Practical Uses

- Measured Signals made up of several (many?) sources
- All overlap in time
- But overlap in frequency may be much less



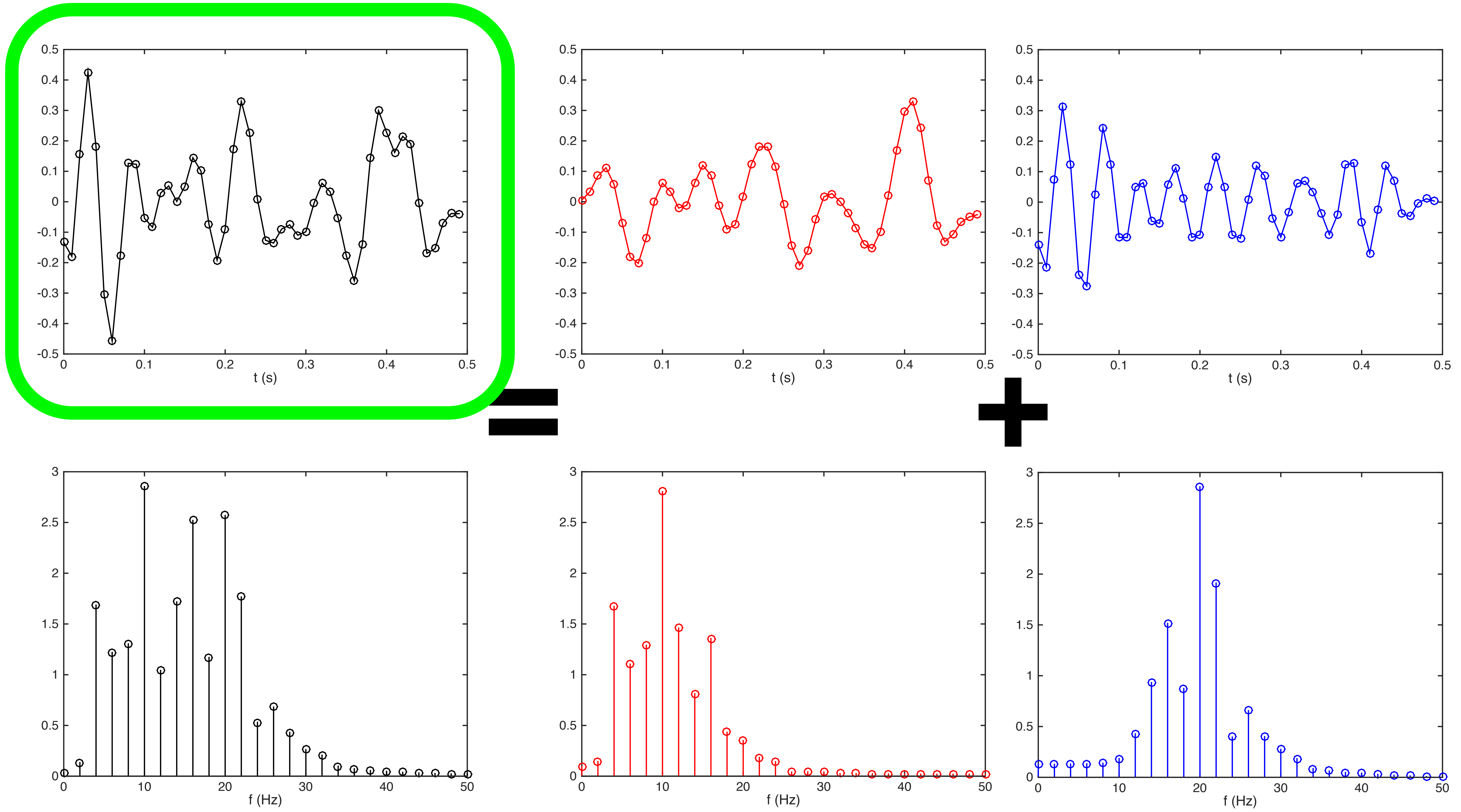
Fourier Transform: Practical Uses

- Measured Signals made up of several (many?) sources
- All overlap in time
- But overlap in frequency may be much less



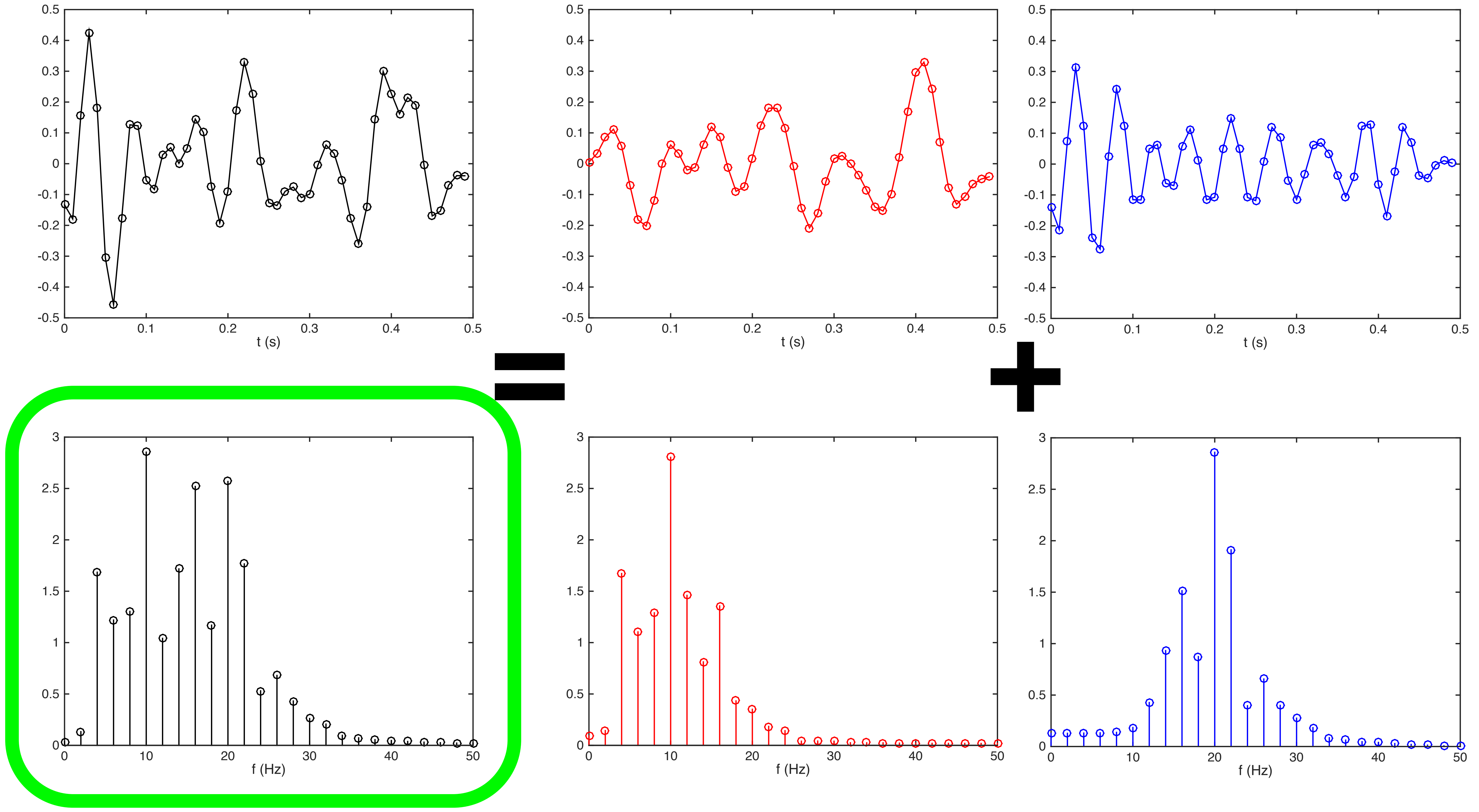
Fourier Transform: Practical Uses

- Measured Signals made up of several (many?) sources
- All overlap in time
- But overlap in frequency may be much less



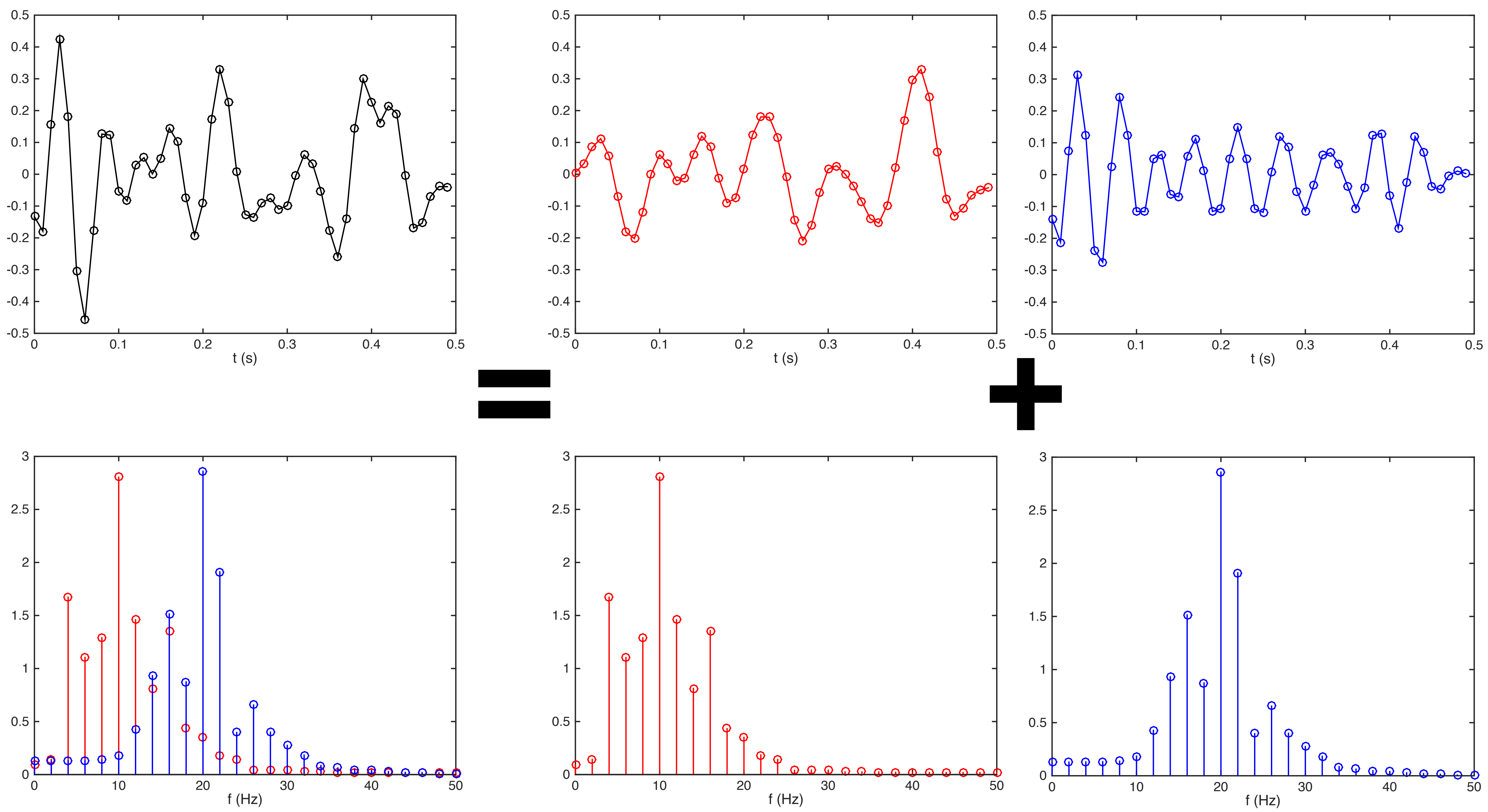
Fourier Transform: Practical Uses

- Measured Signals made up of several (many?) sources
- All overlap in time
- But overlap in frequency may be much less



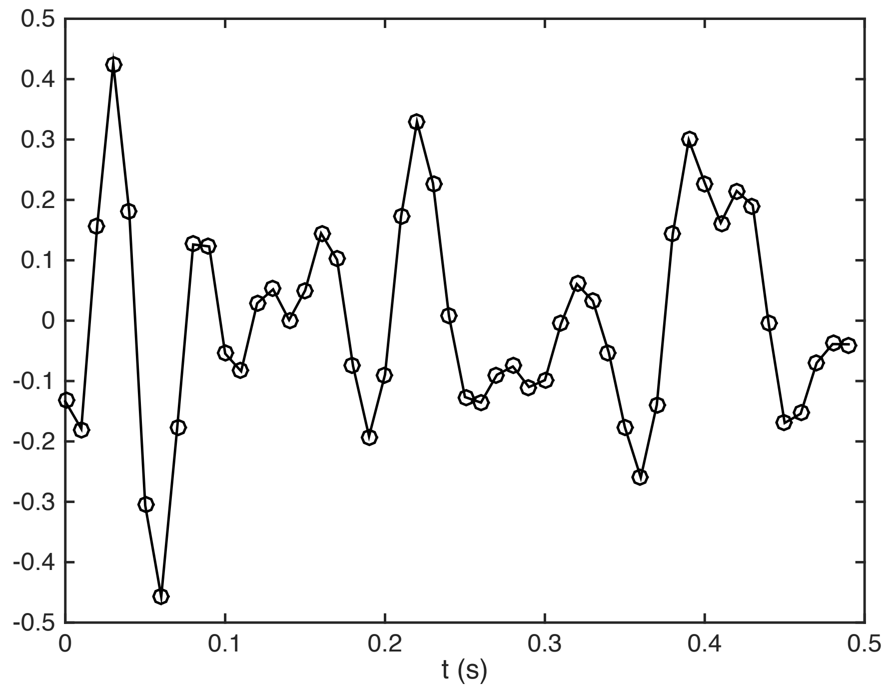
Fourier Transform: Practical Uses

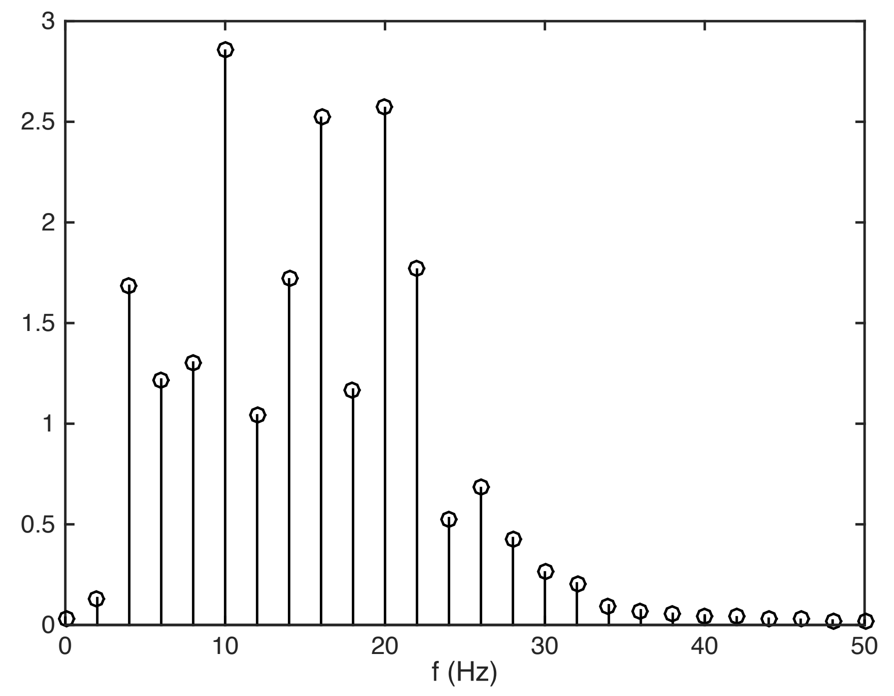
- Measured Signals made up of several (many?) sources
- All overlap in time
- But overlap in frequency may be much less



Fourier Transform: Practical Uses

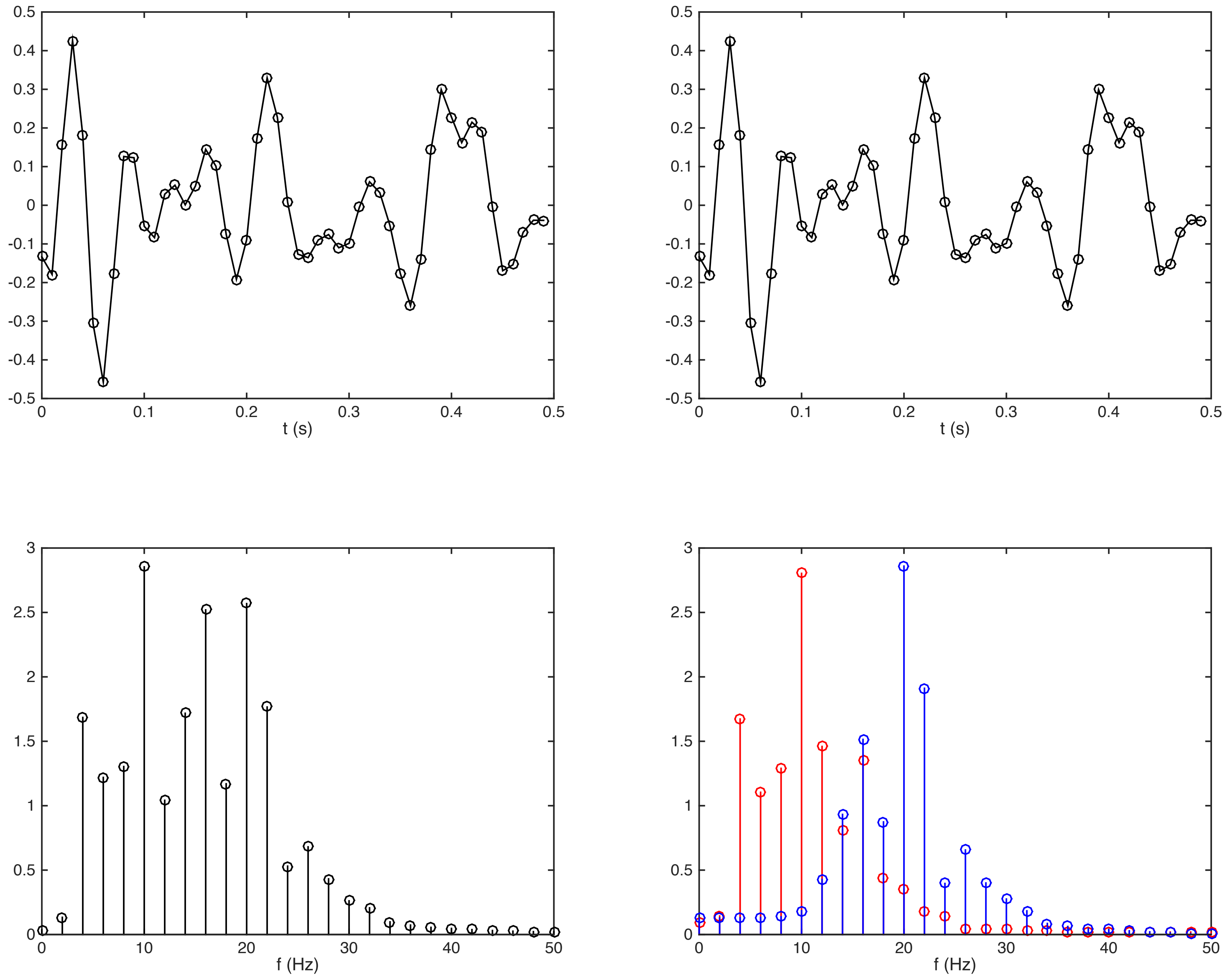
- Measured Signals made up of several (many?) sources
- All overlap in time
- But overlap in frequency may be much less





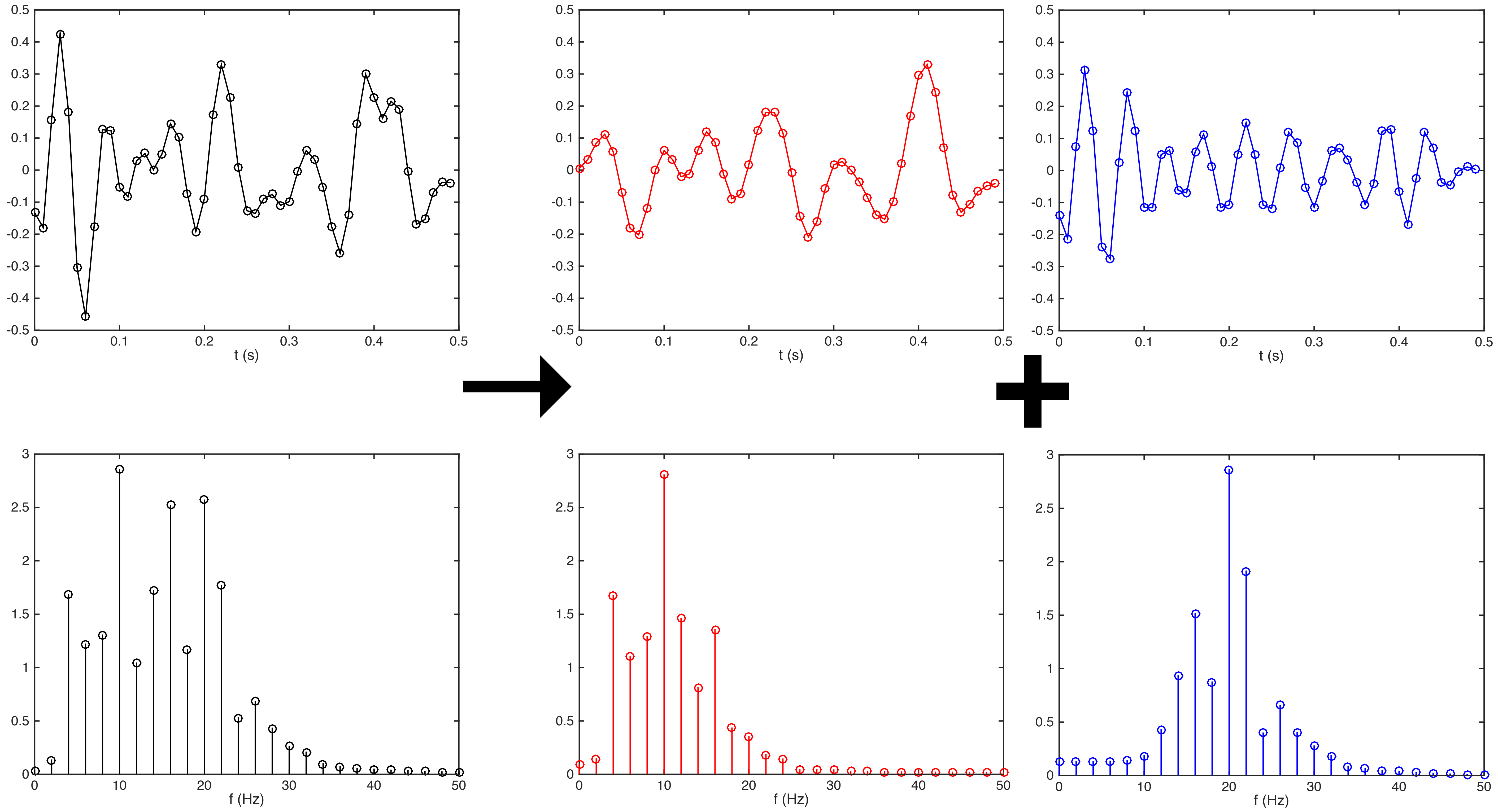
Fourier Transform: Practical Uses

- Measured Signals made up of several (many?) sources
- All overlap in time
- But overlap in frequency may be much less



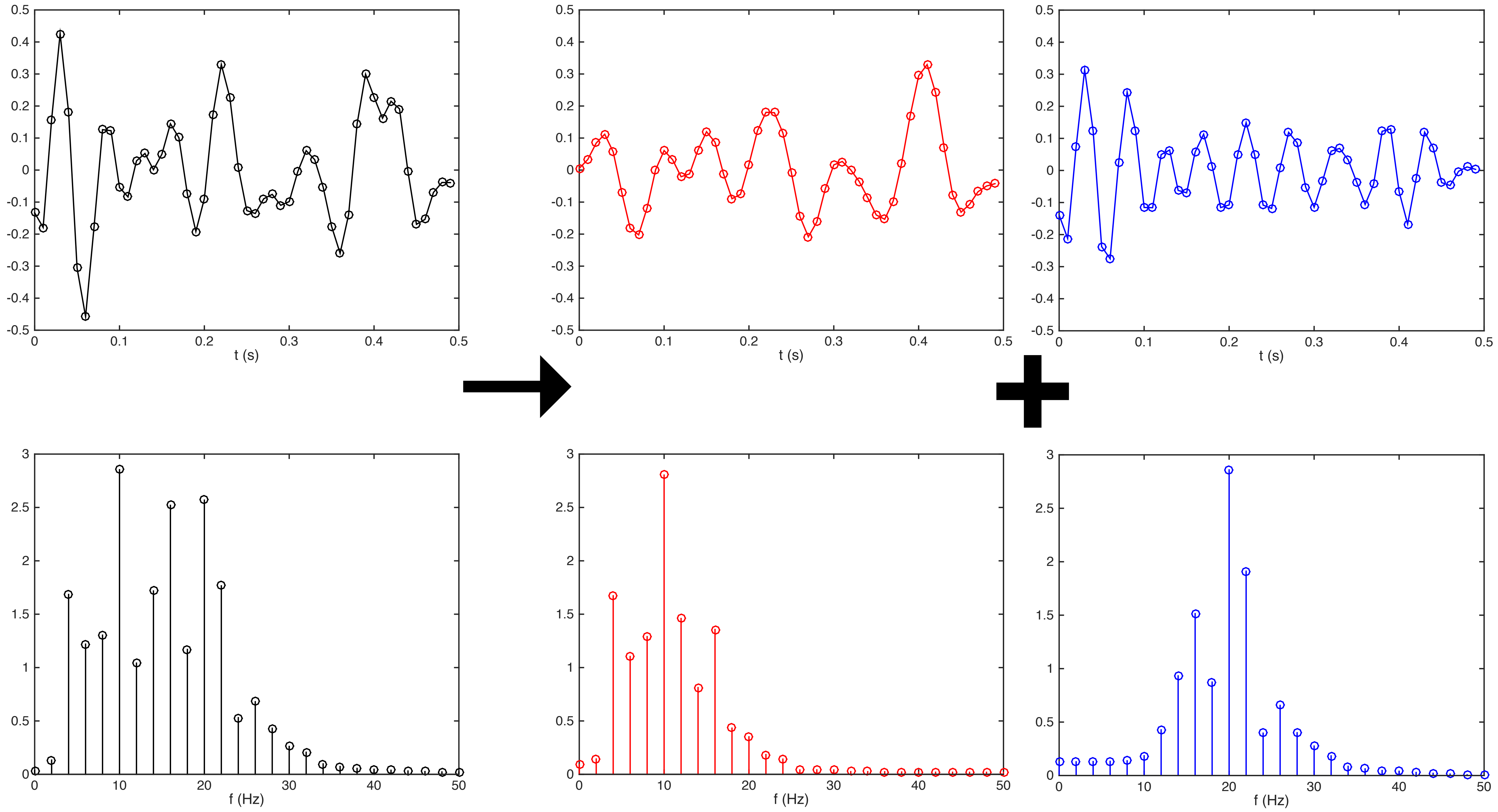
Fourier Transform: Practical Uses

- Measured Signals made up of several (many?) sources
- All overlap in time
- But overlap in frequency may be much less



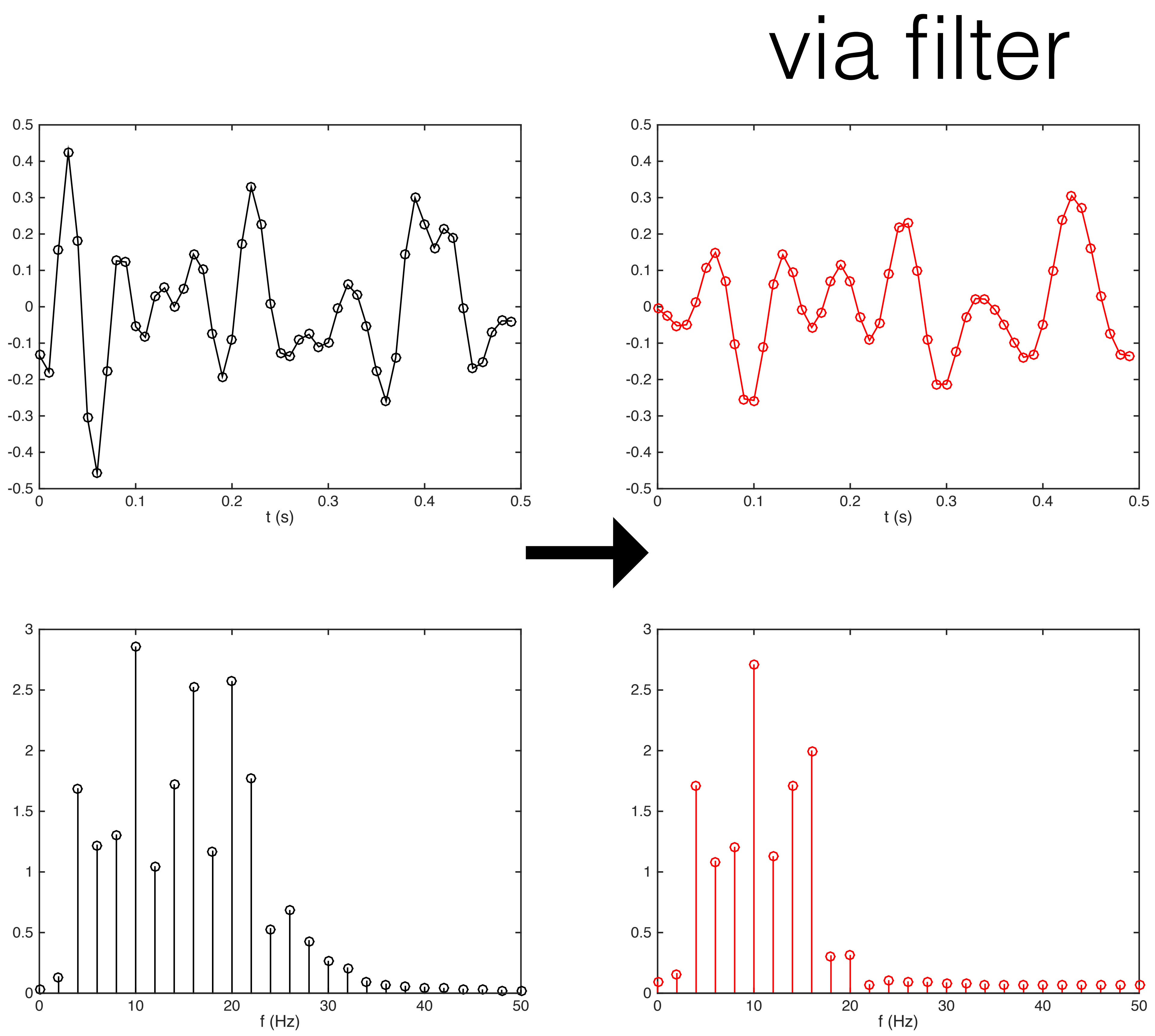
Fourier Transform: Practical Uses

- Measured Signals made up of several (many?) sources
- All overlap in time
- But overlap in frequency may be much less
- **Can *filter* measured (mixed) signal to "recover" underlying source signal**



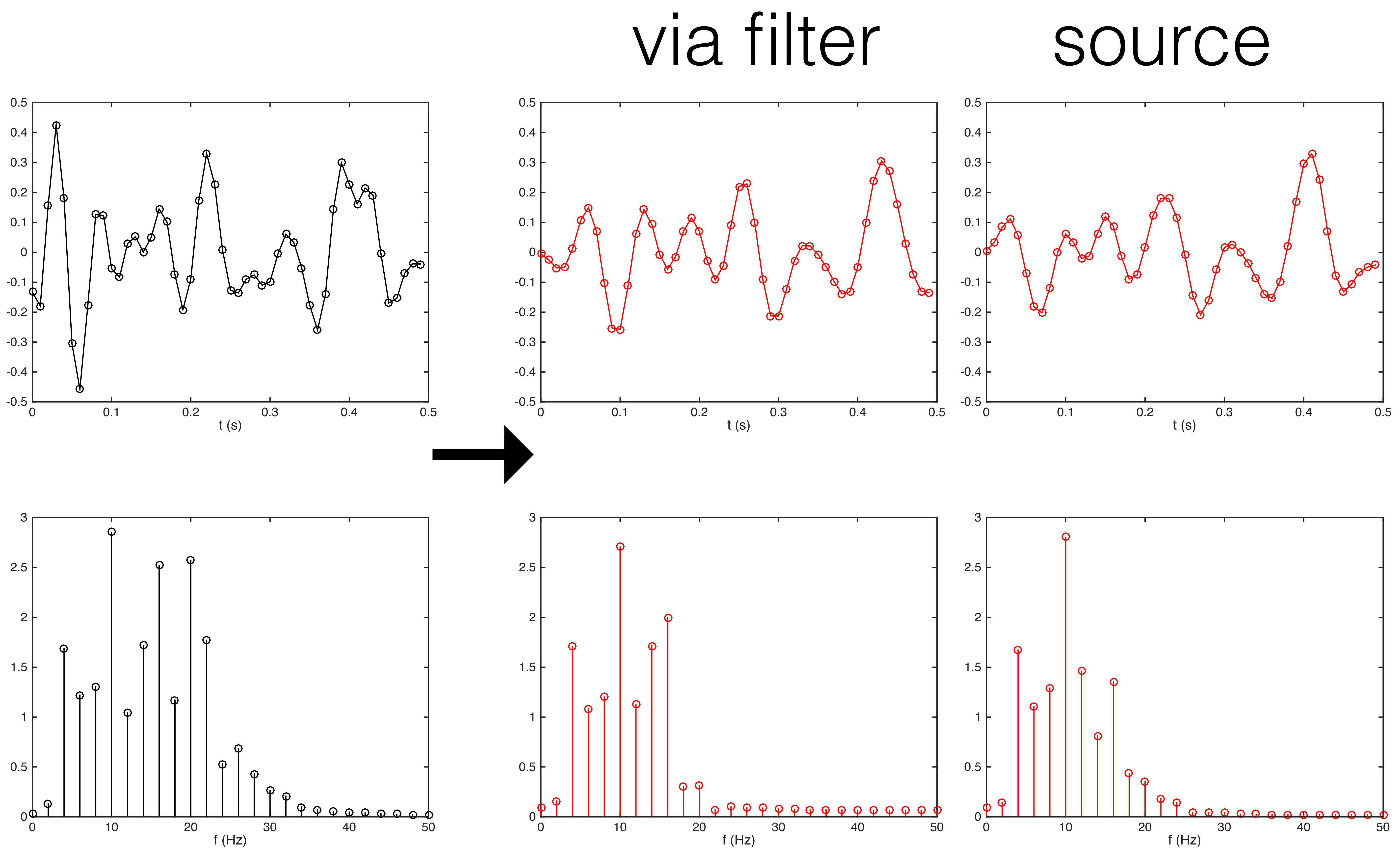
Fourier Transform: Filtering

- Measured Signals made up of several (many?) sources
- All overlap in time
- But overlap in frequency may be much less
- **Can *filter* measured (mixed) signal to "recover" underlying source signal**



Fourier Transform: Filtering

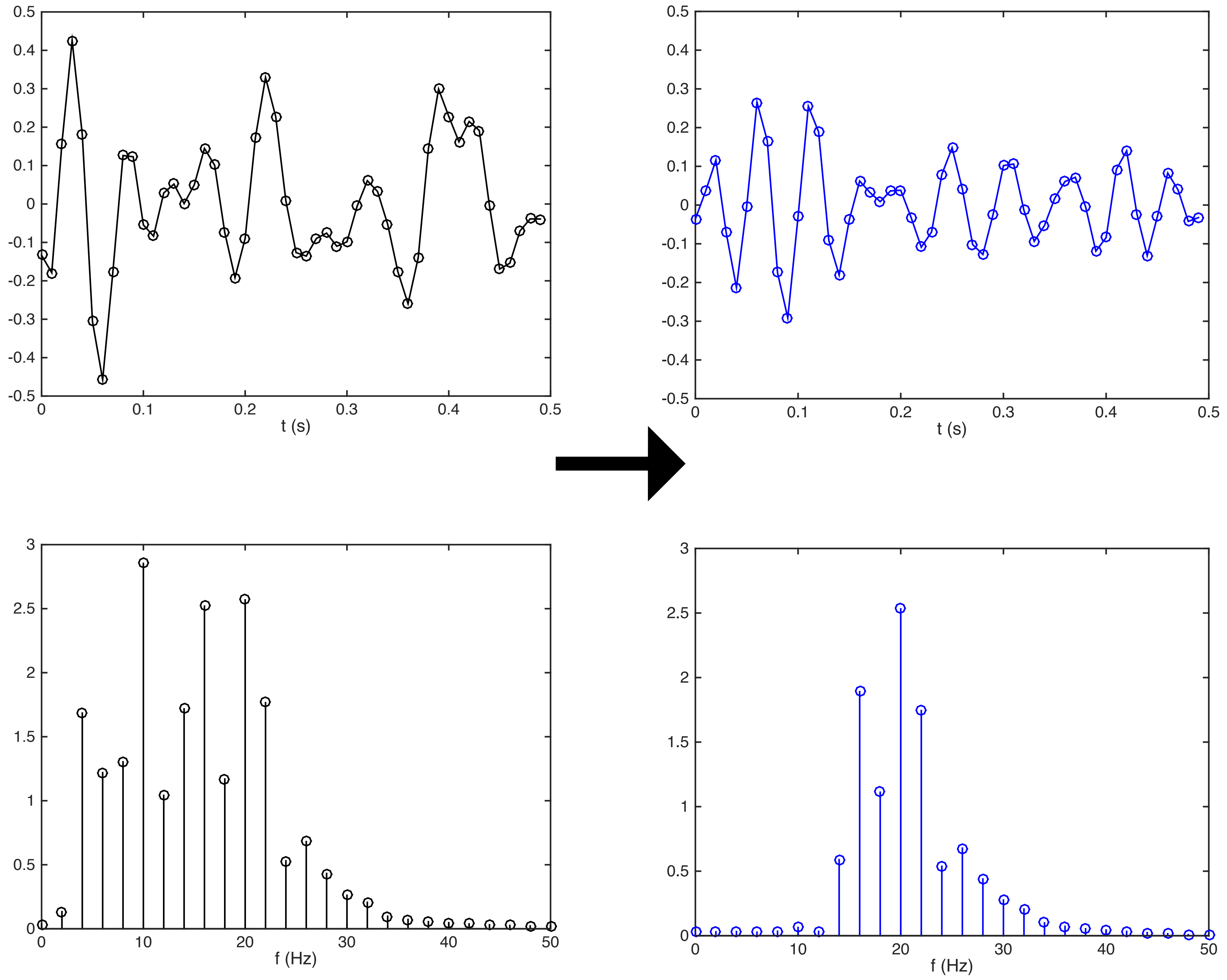
- Measured Signals made up of several (many?) sources
- All overlap in time
- But overlap in frequency may be much less
- **Can *filter* measured (mixed) signal to "recover" underlying source signal**



Fourier Transform: Filtering

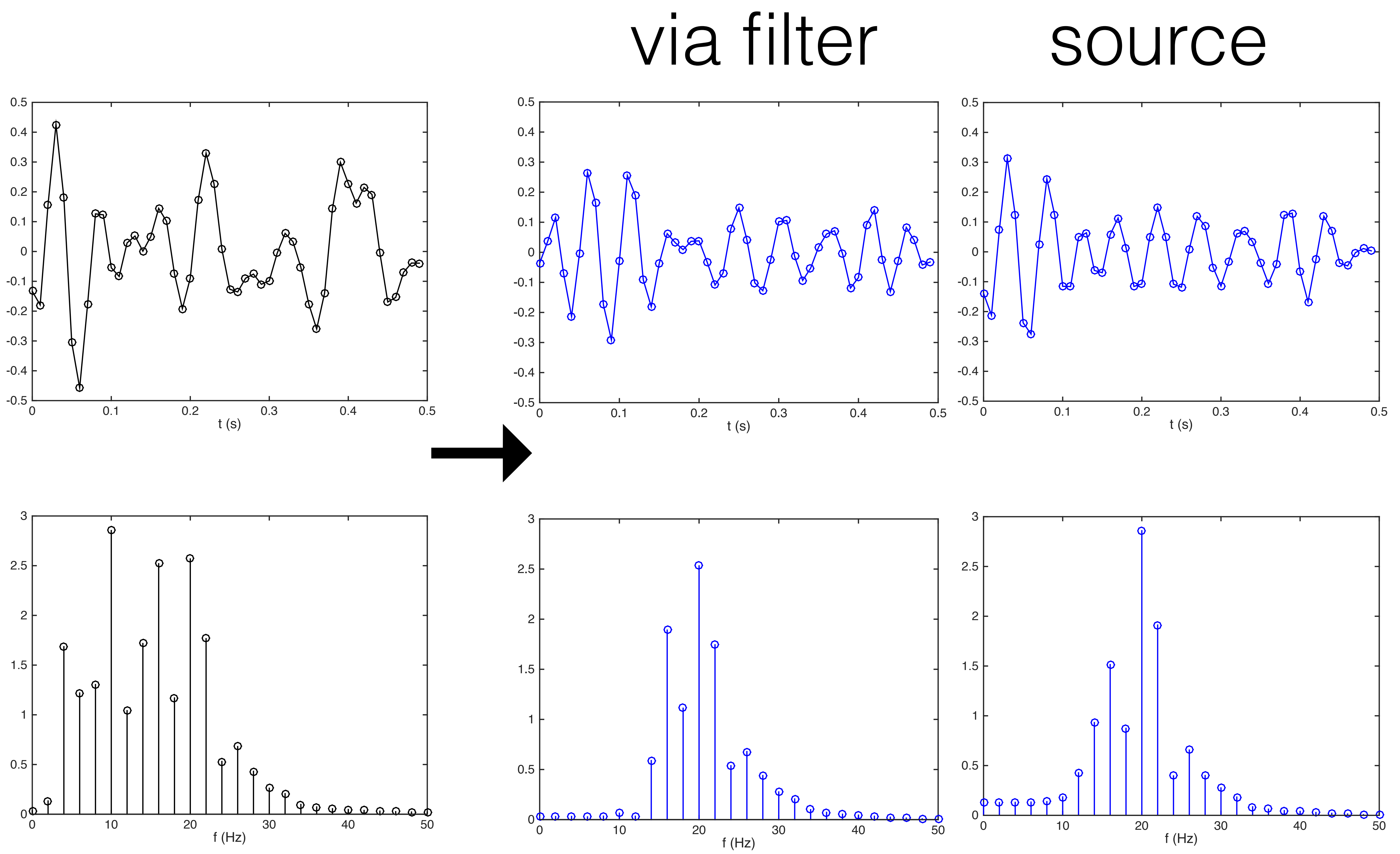
- Measured Signals made up of several (many?) sources
- All overlap in time
- But overlap in frequency may be much less
- **Can *filter* measured (mixed) signal to “recover” underlying source signal**

via filter



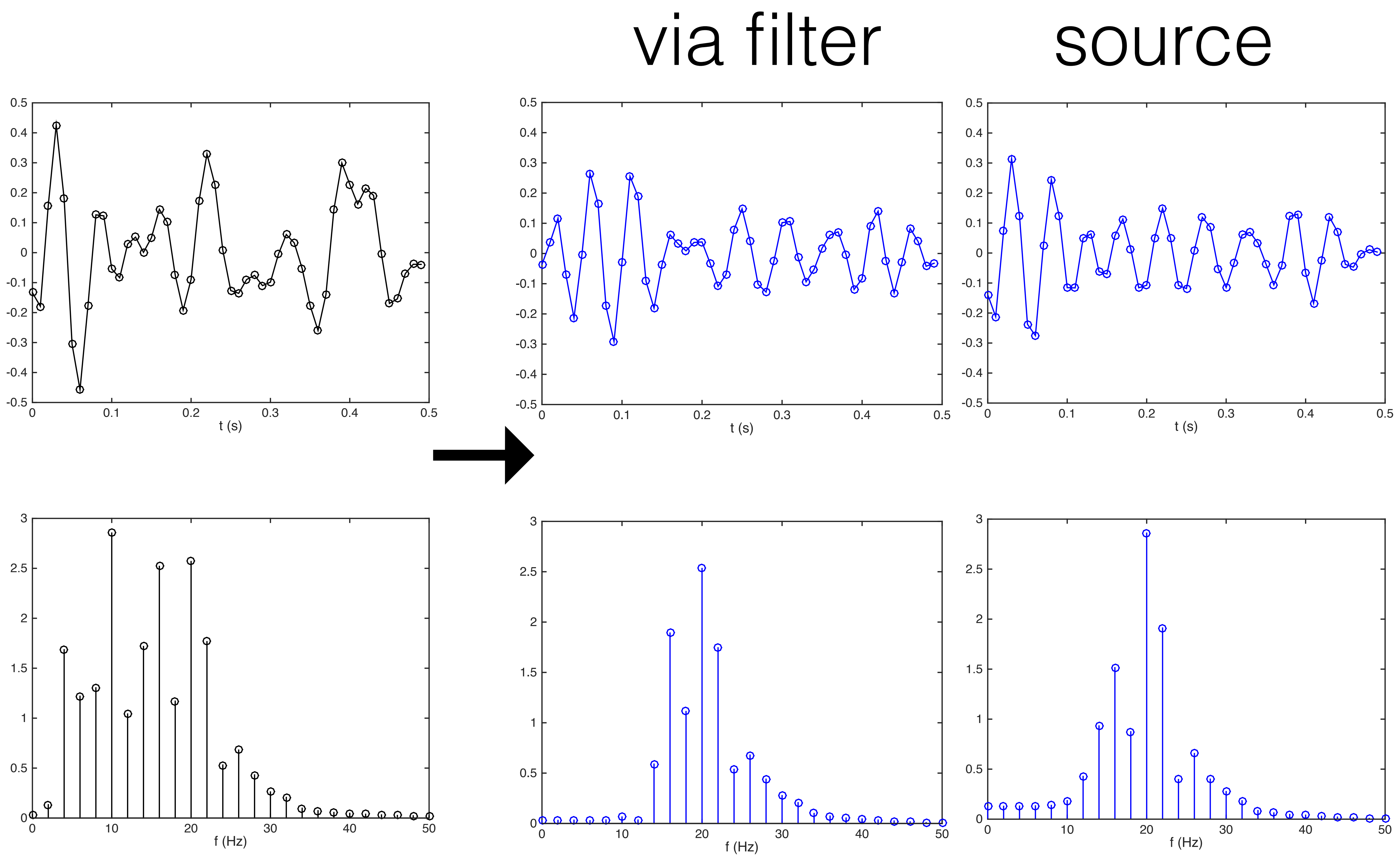
Fourier Transform: Filtering

- Measured Signals made up of several (many?) sources
- All overlap in time
- But overlap in frequency may be much less
- **Can *filter* measured (mixed) signal to "recover" underlying source signal**



Break for Computer Lab Exercise 2

- Measured Signals made up of several (many?) sources
- All overlap in time
- But overlap in frequency may be much less
- **Can *filter* measured (mixed) signal to “recover” underlying source signal**



Outline

- Fourier Transform: *Why It's Useful*, and *What it Can/Cannot Do For You*
- Filters: *What They Do*, and *How They Do It*
- Filters: *Why So Many Different Kinds? Which Should I Use and When*?
- Grab Bag:
  - *Use Causal Filters; Windowing is Good*

Outline

- Fourier Transform: *Why It's Useful*, and *What it Can/Cannot Do For You*
- Filters: *What They Do*, and *How They Do It*
- Filters: *Why So Many Different Kinds? Which Should I Use and When*?
- Grab Bag:
  - *Use Causal Filters; Windowing is Good*

Filters: Frequency Selectivity

- *Frequency Selective* Filters

Low Pass

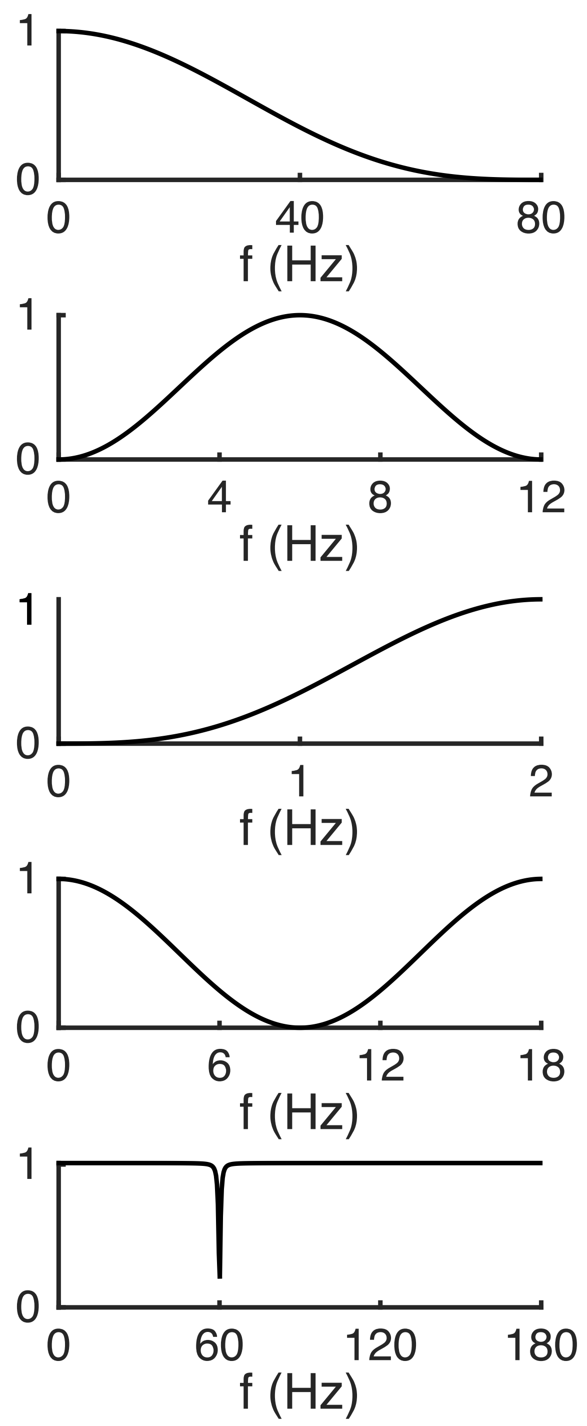
Band Pass

High Pass

Band Stop

Notch

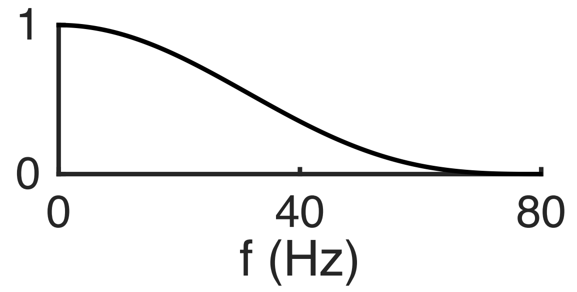
and more…



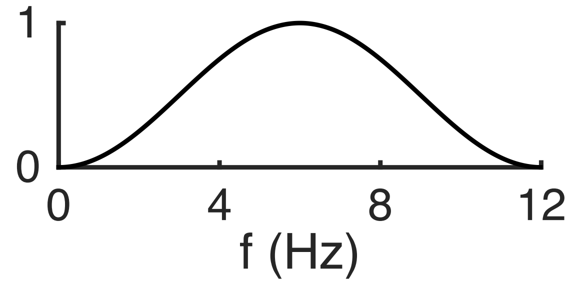
Filters: Frequency Selectivity

- *Frequency Selective* Filters

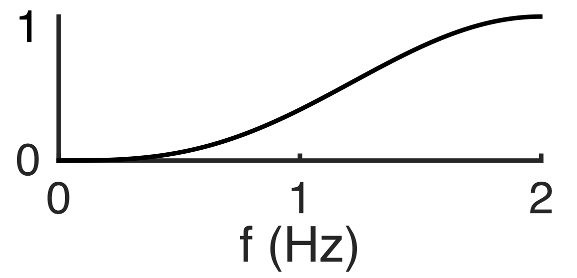
Low Pass



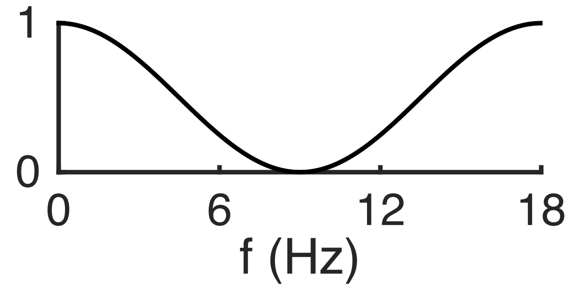
Band Pass



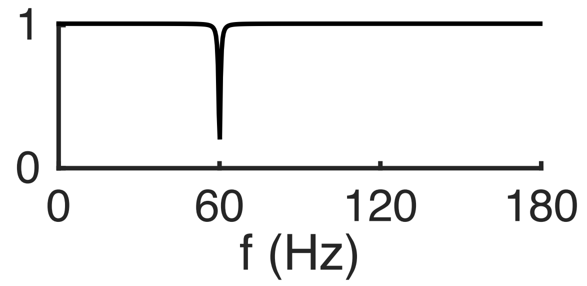
High Pass



Band Stop

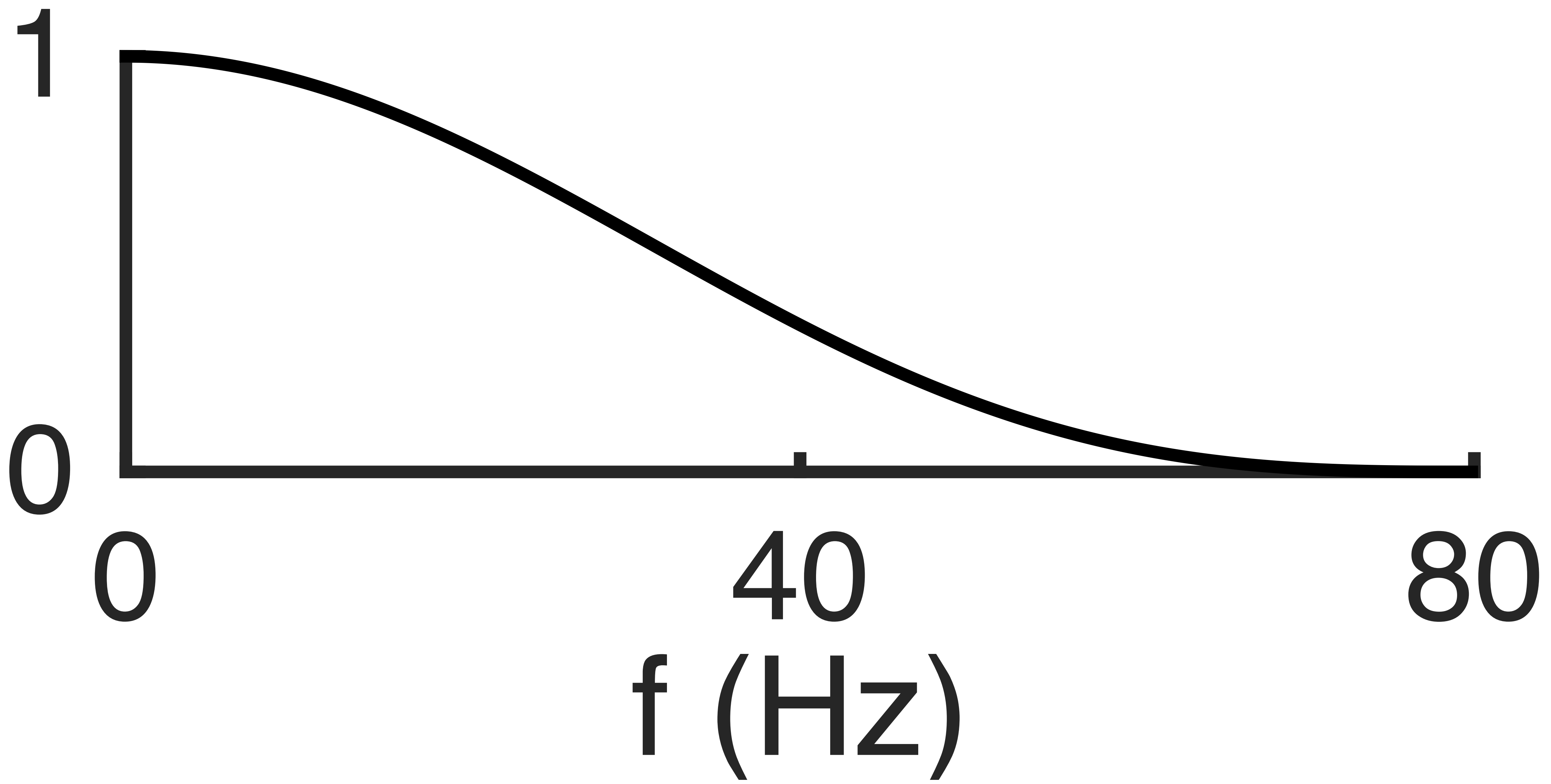


Notch



and more…

Low Pass



Filters: Frequency Selectivity

- *Frequency Selective* Filters

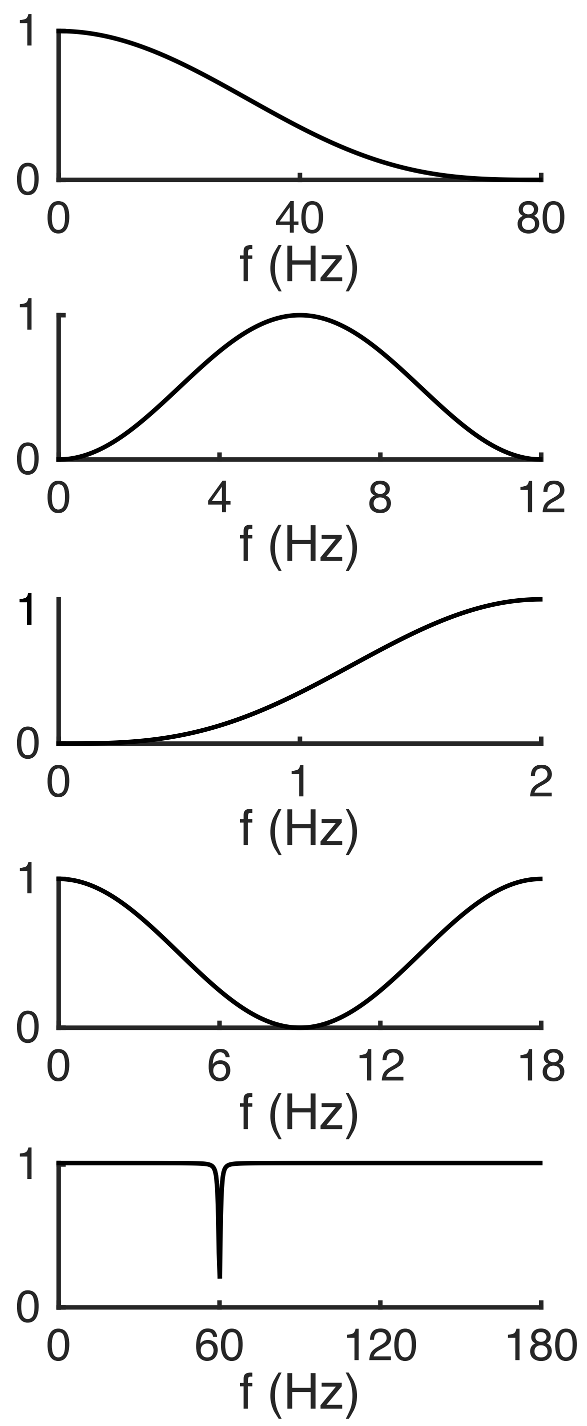
Low Pass

Band Pass

High Pass

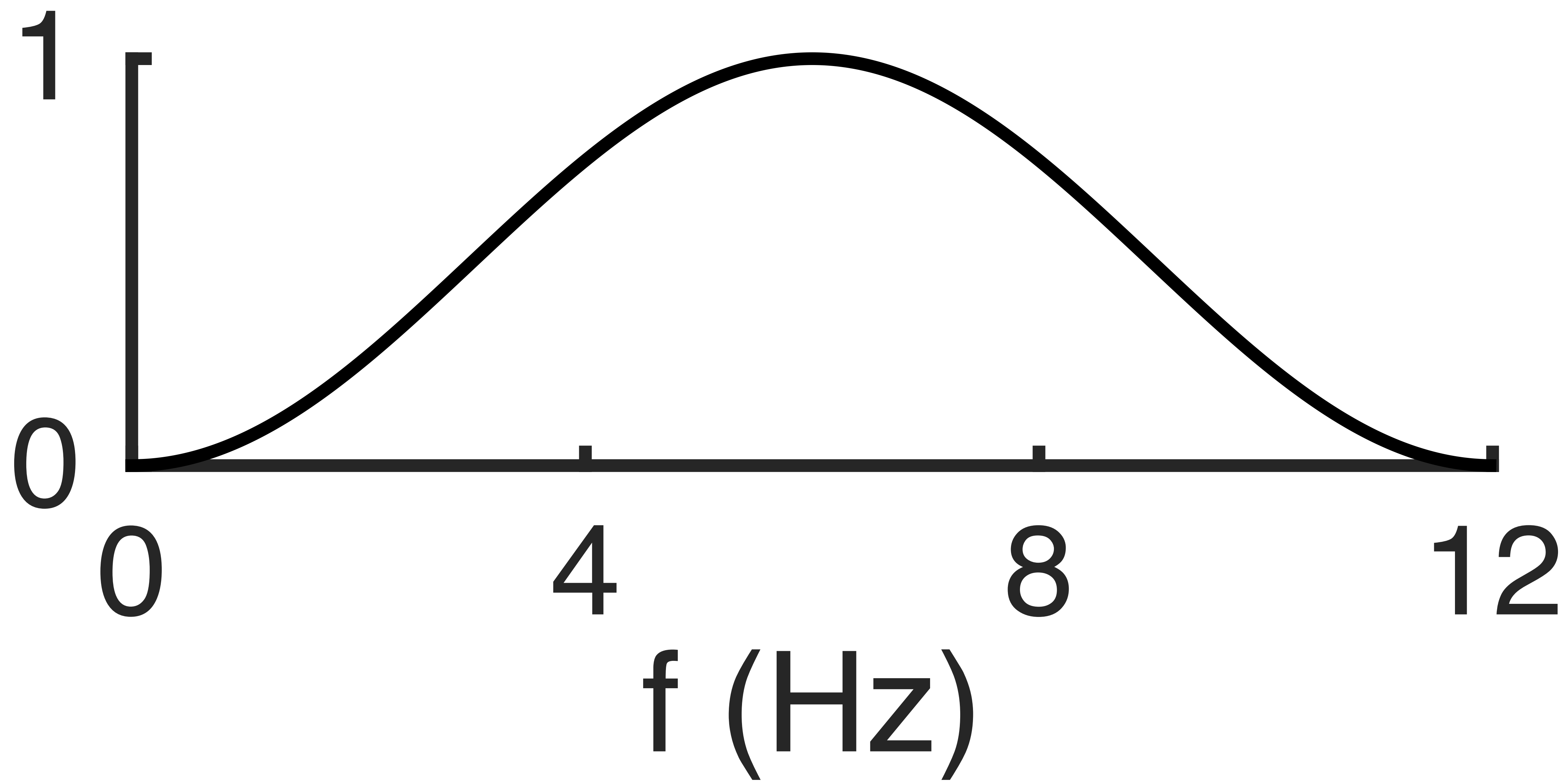
Band Stop

Notch



and more…

Band Pass



Filters: Frequency Selectivity

- *Frequency Selective* Filters

Low Pass

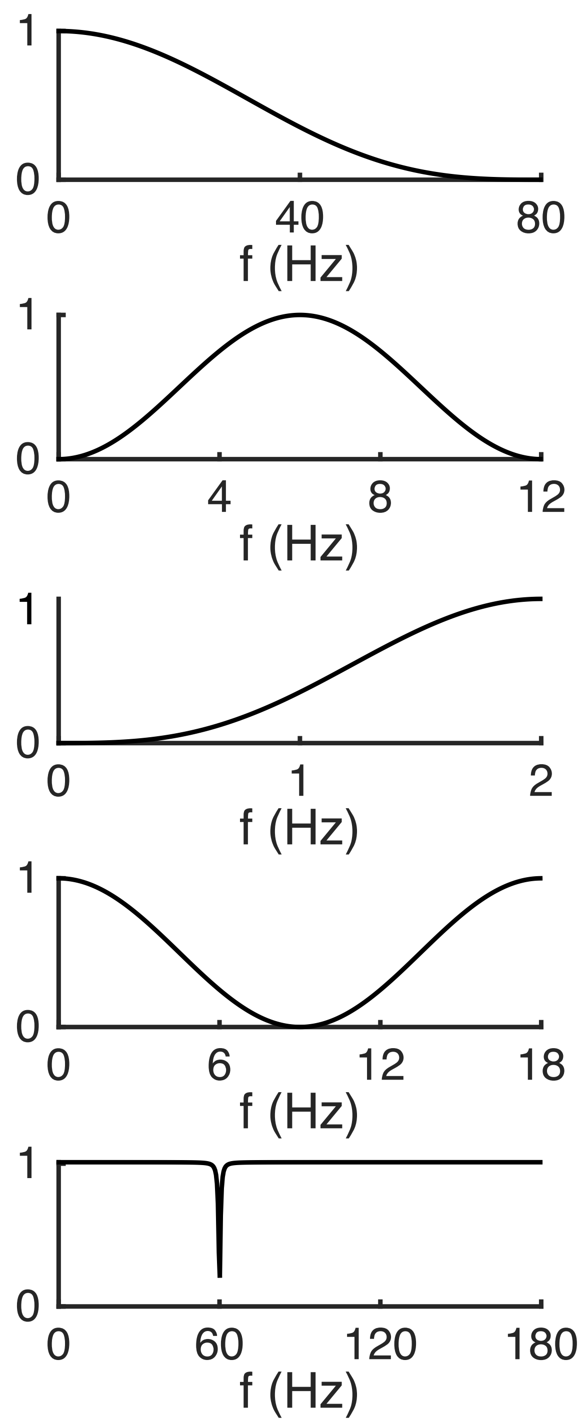
Band Pass

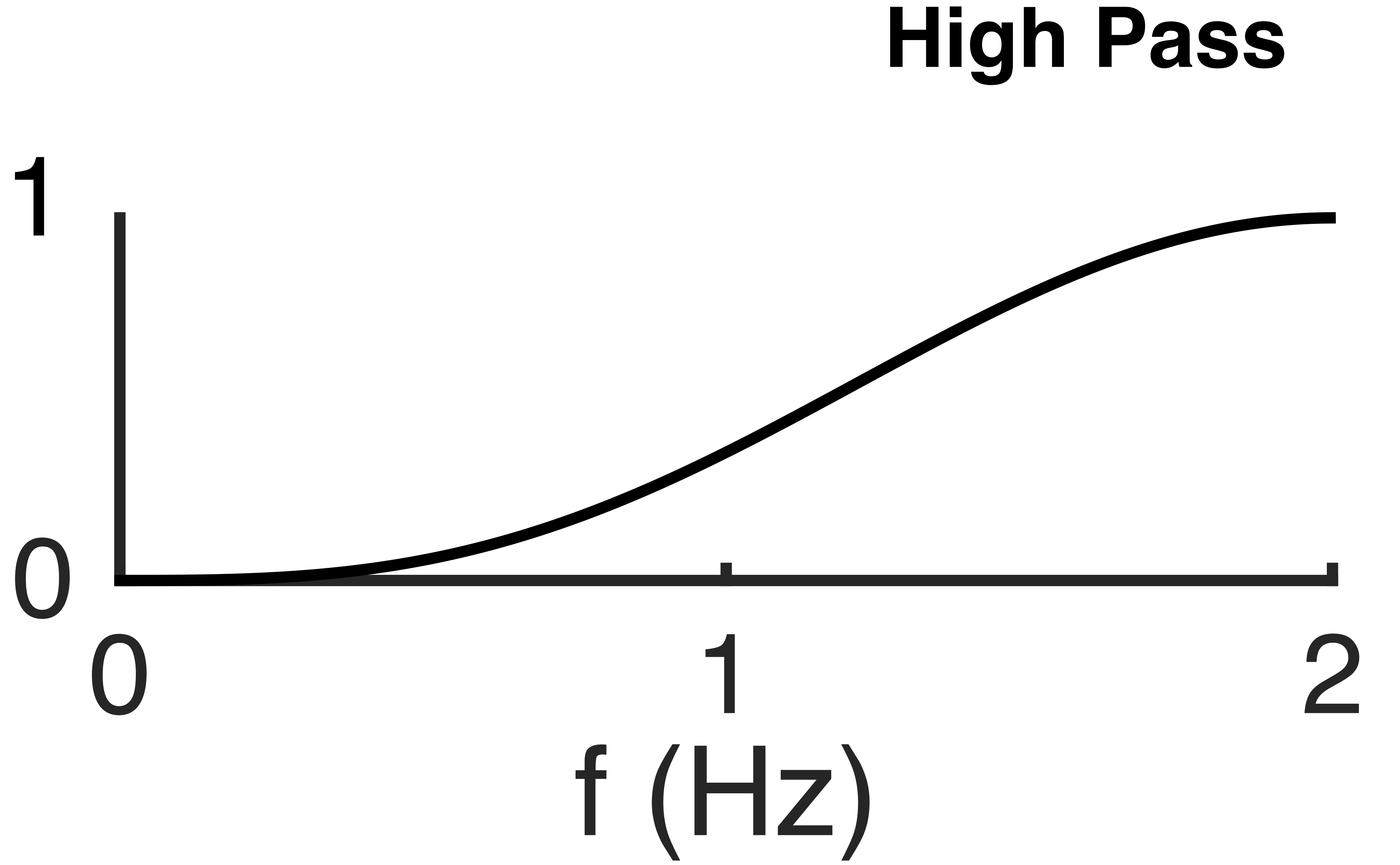
High Pass

Band Stop

Notch

and more…





Filters: Frequency Selectivity

- *Frequency Selective* Filters

Low Pass

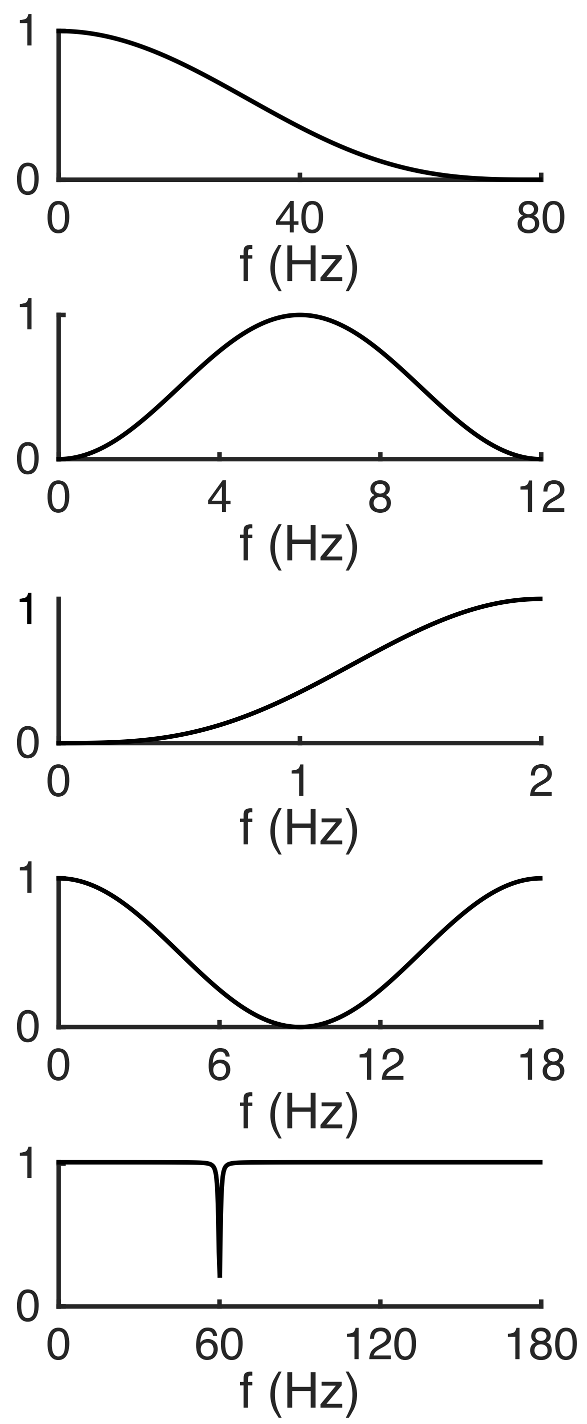
Band Pass

High Pass

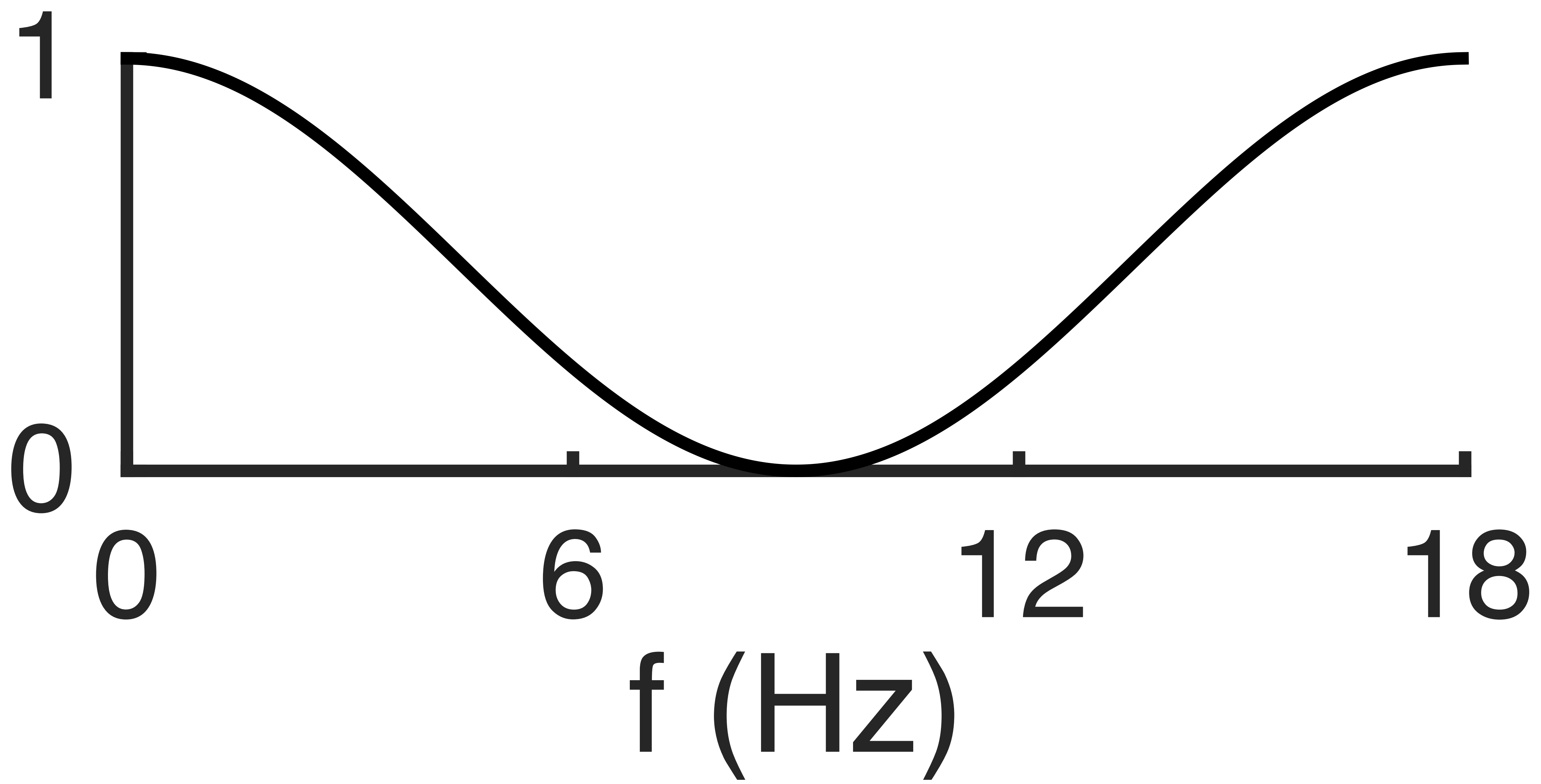
Band Stop

Notch

and more…



Band Stop



Filters: Frequency Selectivity

- *Frequency Selective* Filters

Low Pass

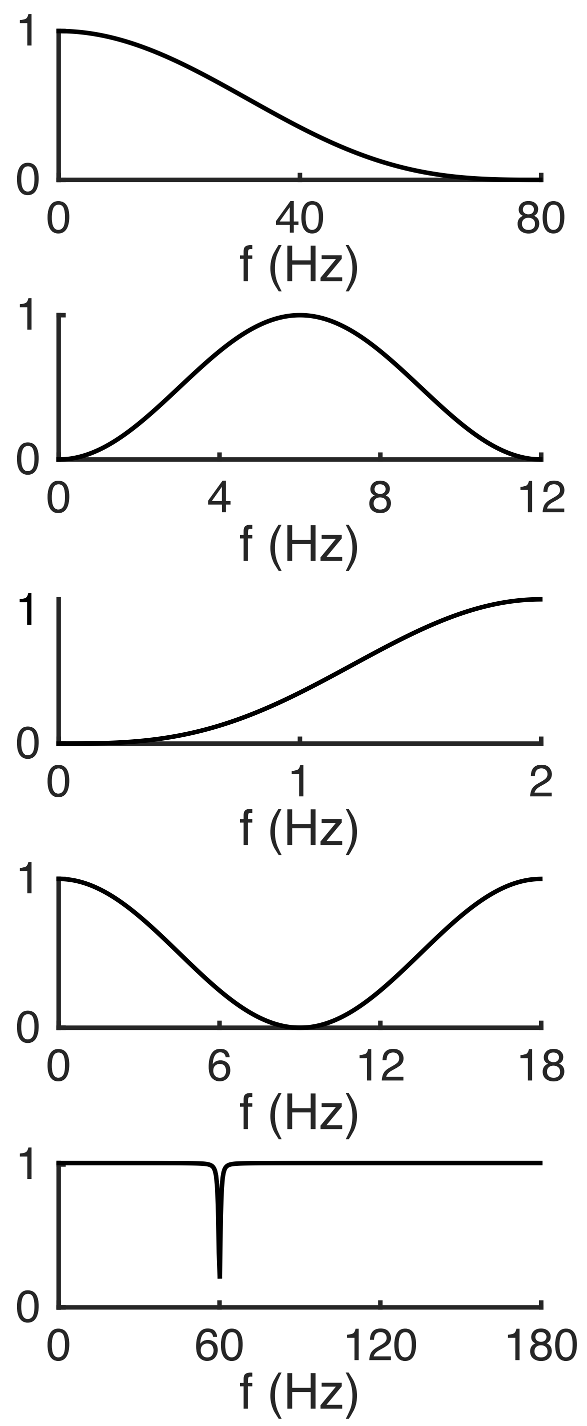
Band Pass

High Pass

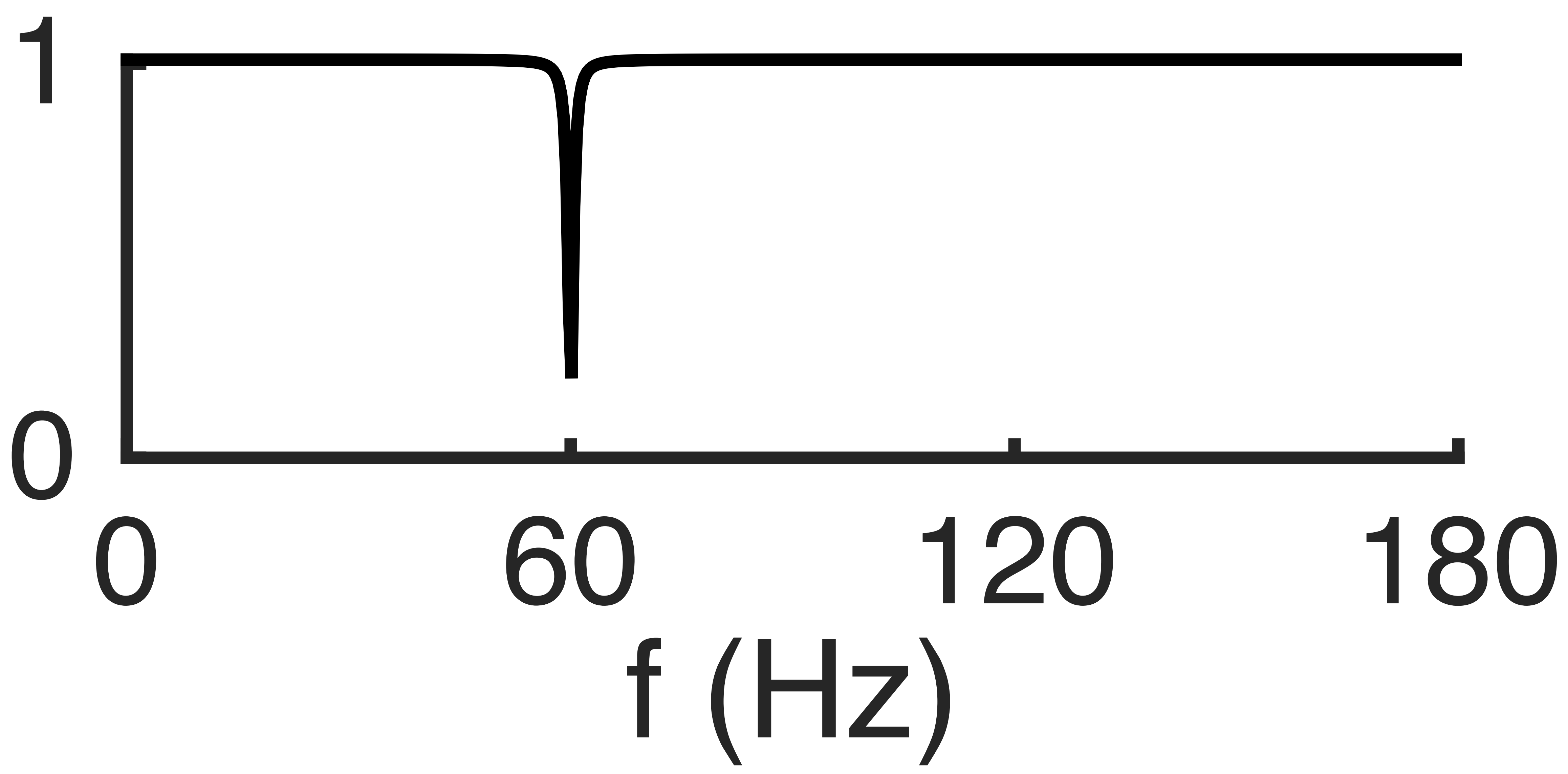
Band Stop

Notch

and more…



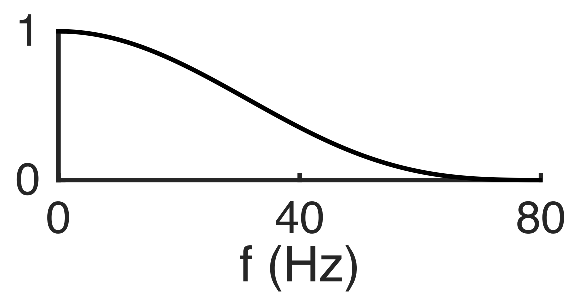
Notch



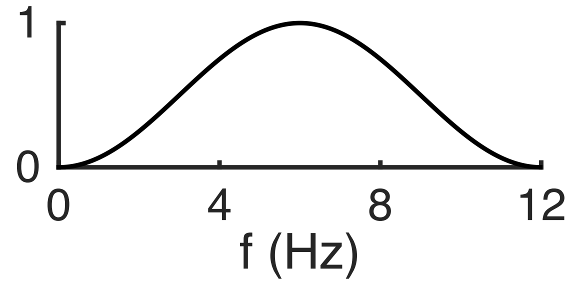
Filters: Frequency Selectivity

- *Frequency Selective* Filters

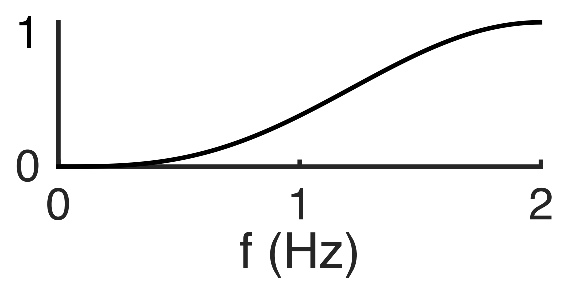
Low Pass



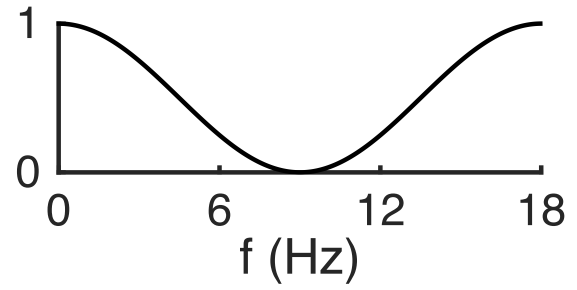
Band Pass



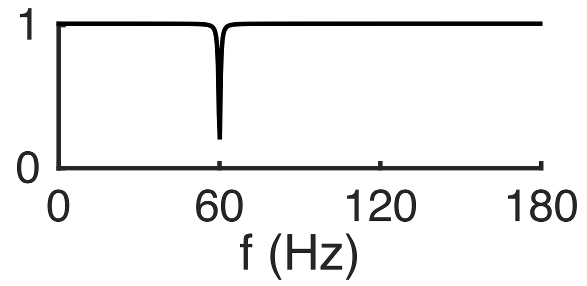
High Pass



Band Stop



Notch

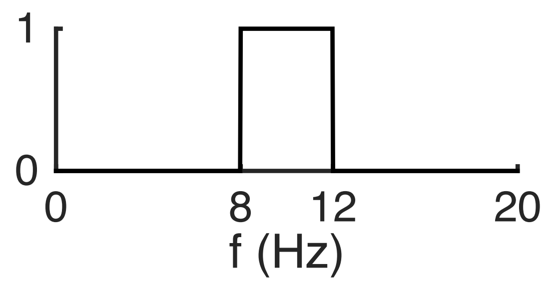


and more…

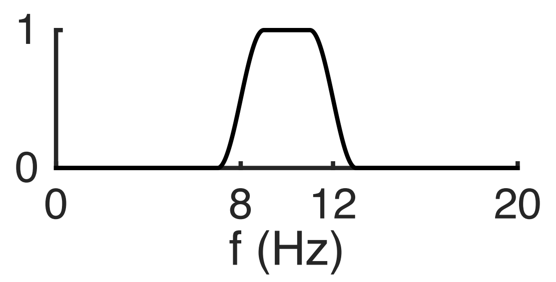
Filters: How Selective?

- *How sharp a transition?*

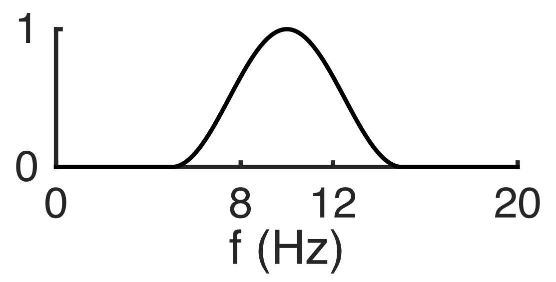
"Ideal" Filter



Sharp Transition



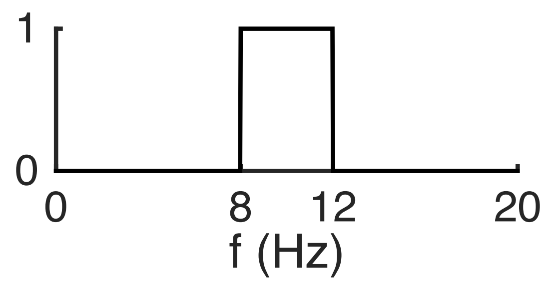
Soft Transition



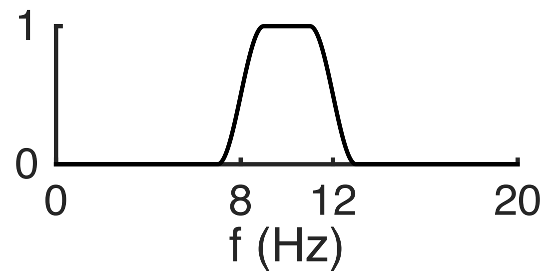
Filters: How Selective?

- *How sharp a transition?*

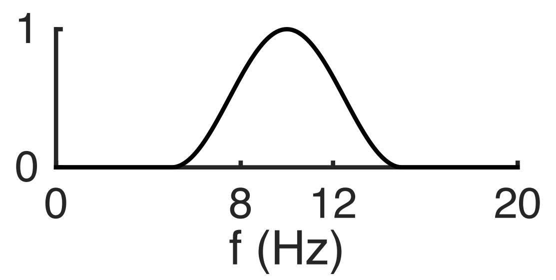
“Ideal” Filter



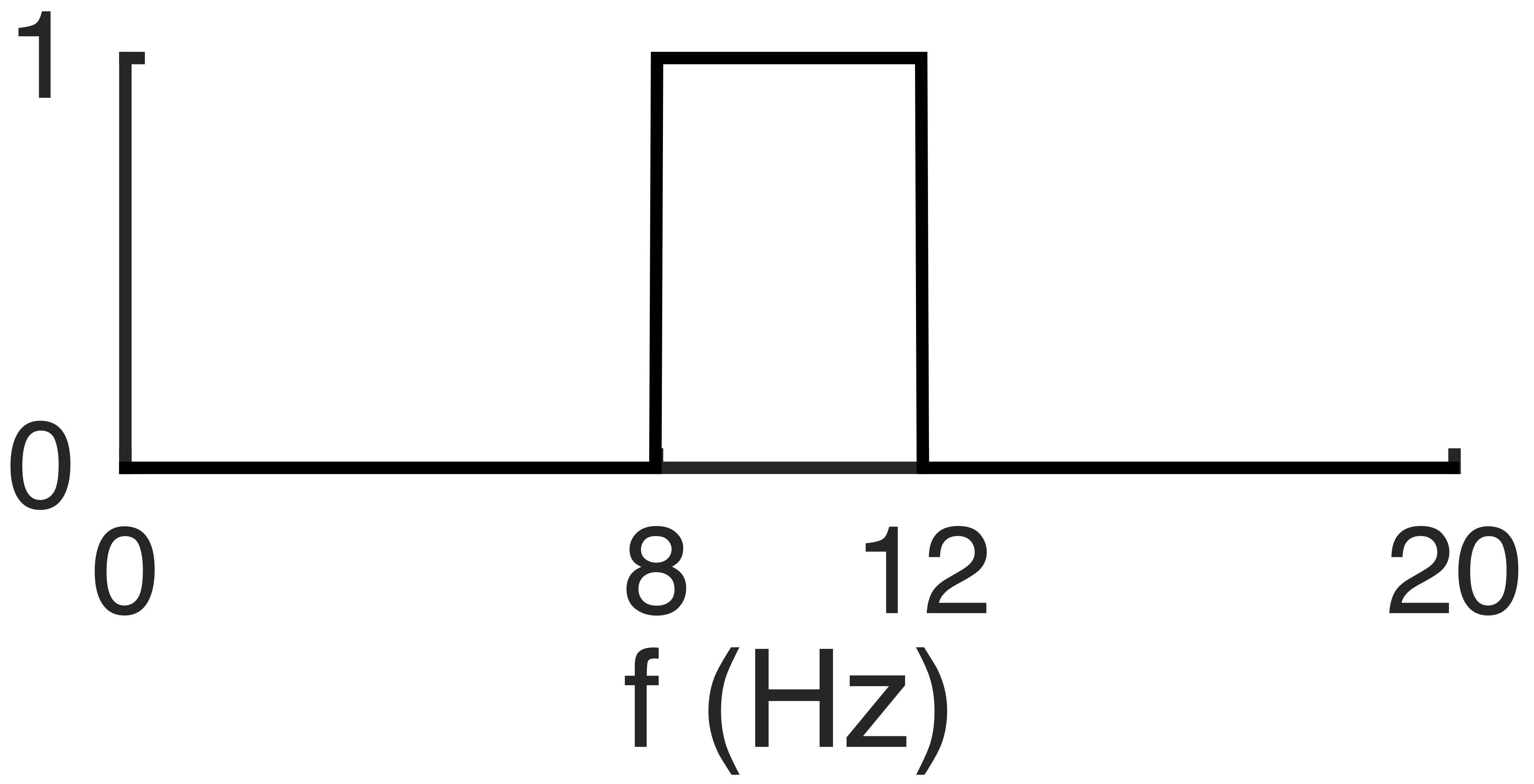
Sharp Transition



Soft Transition



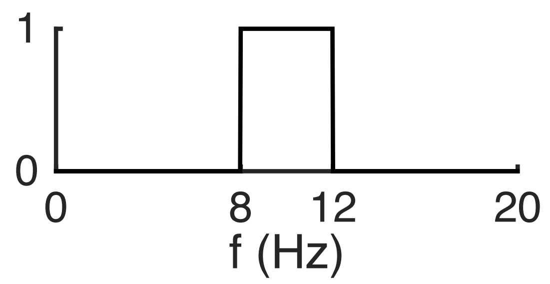
“Ideal” Filter



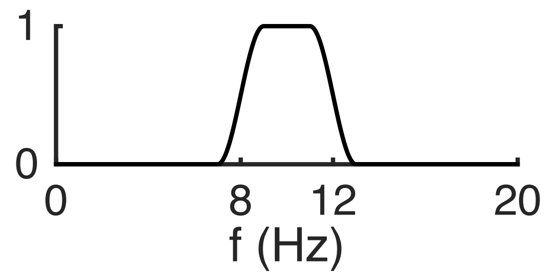
Filters: How Selective?

- *How sharp a transition?*

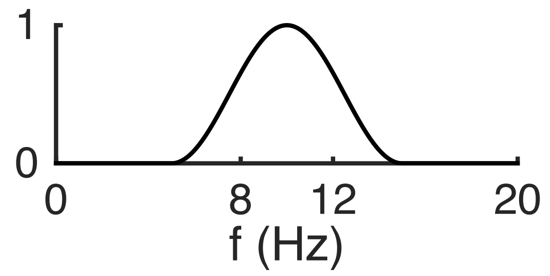
"Ideal" Filter

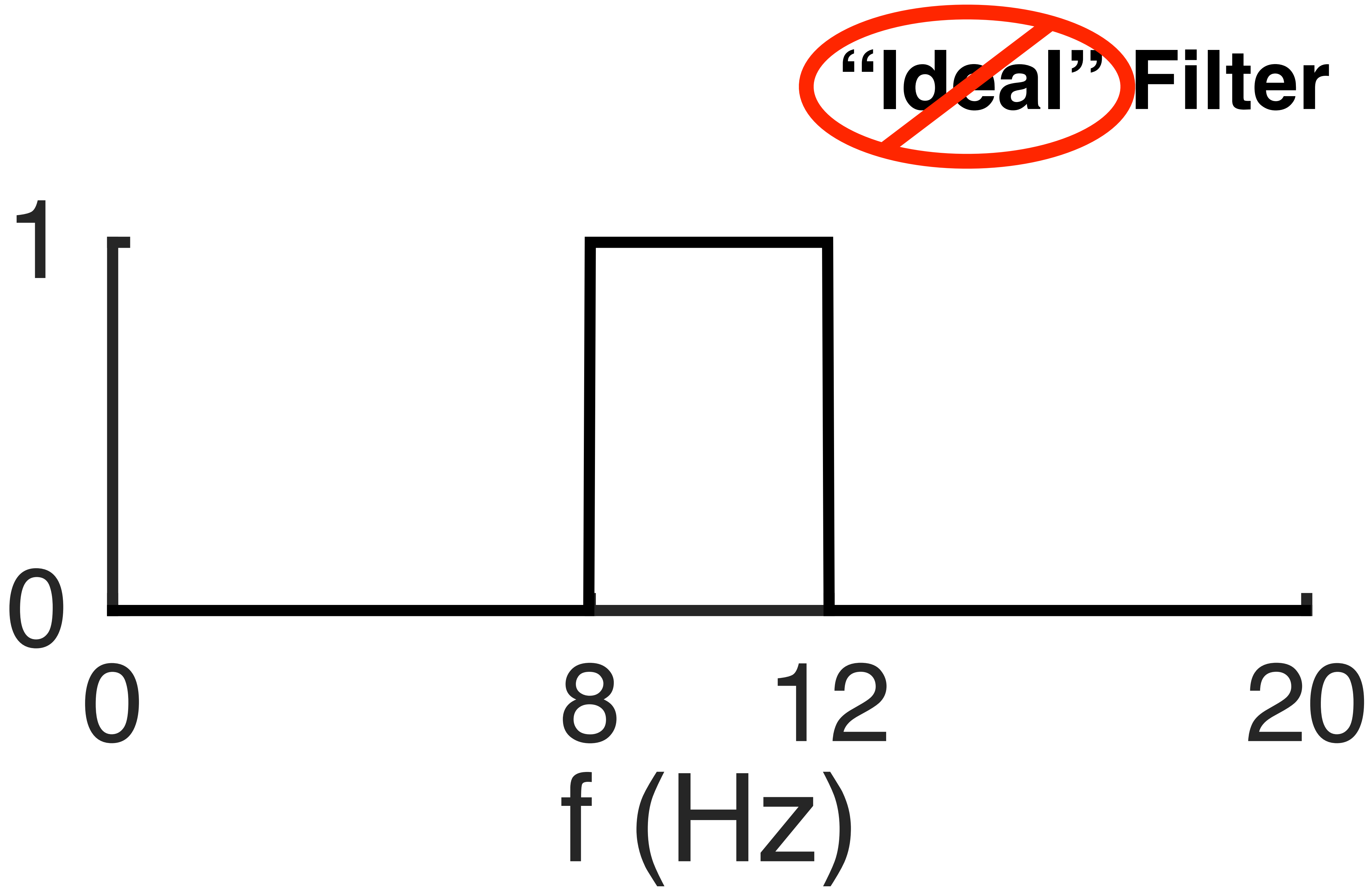


Sharp Transition



Soft Transition

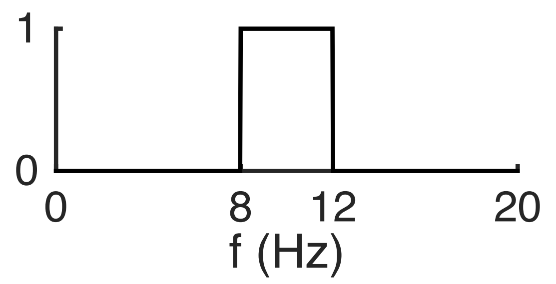




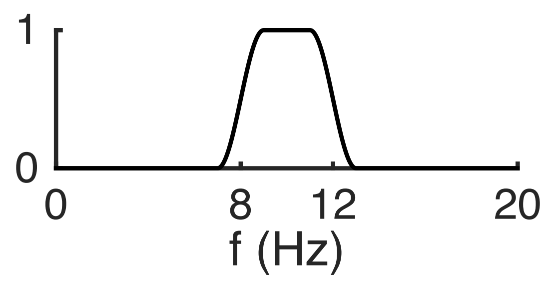
Filters: How Selective?

- *How sharp a transition?*

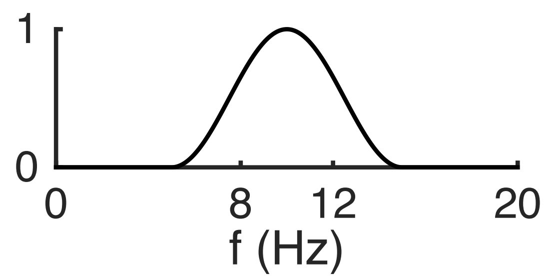
“Ideal” Filter



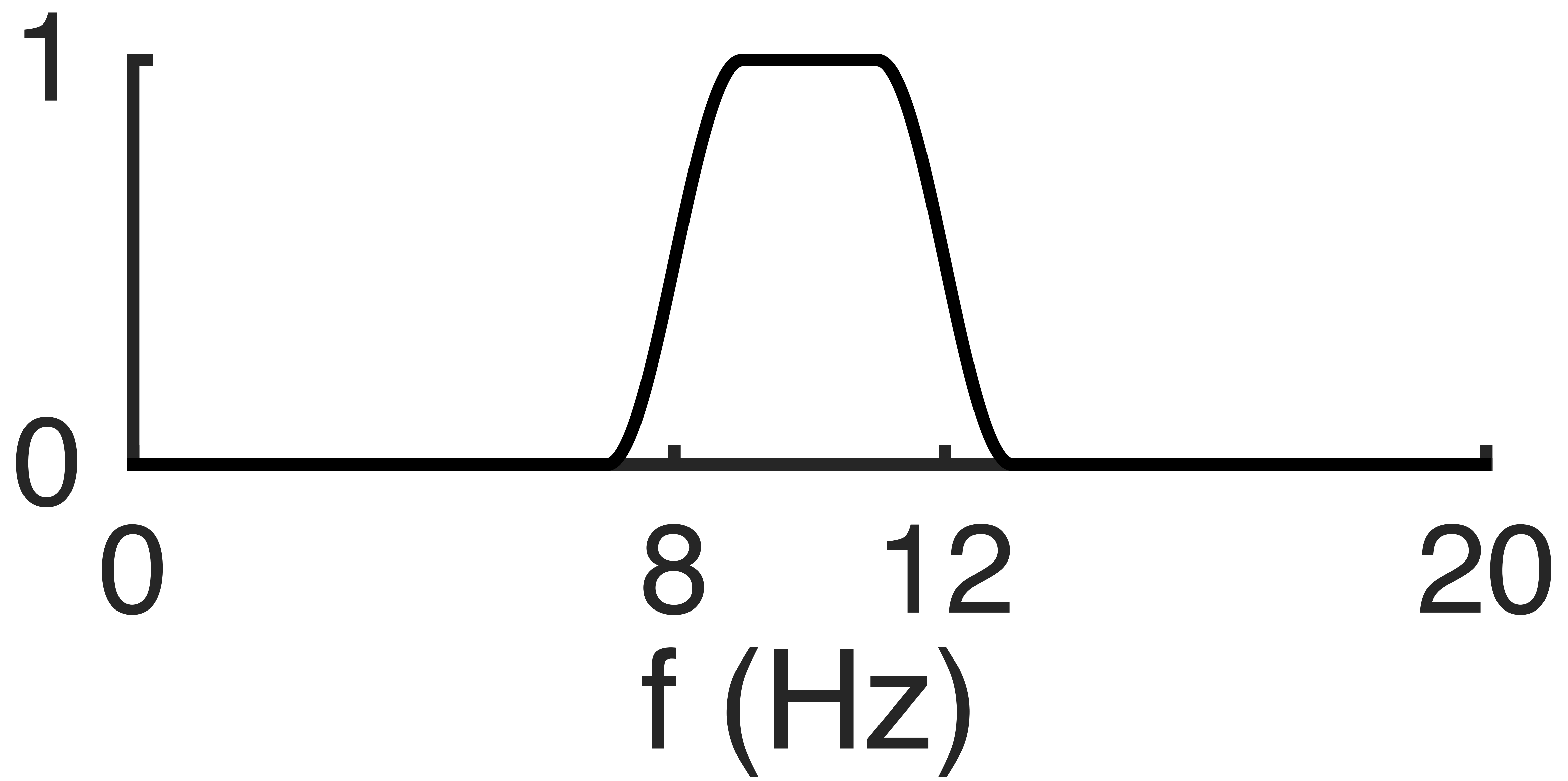
Sharp Transition



Soft Transition



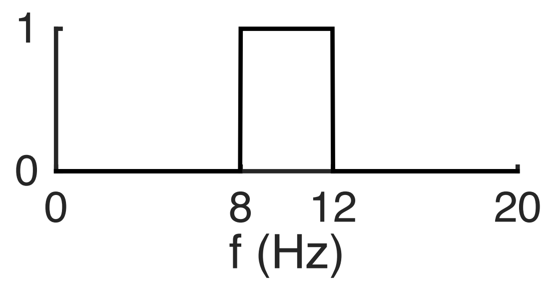
Sharp Transition



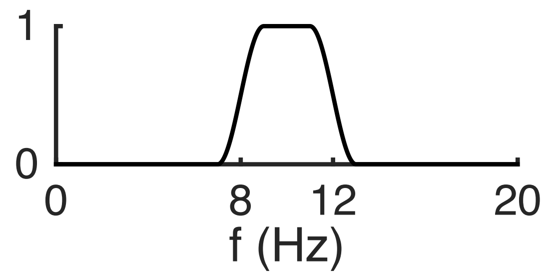
Filters: How Selective?

- *How sharp a transition?*

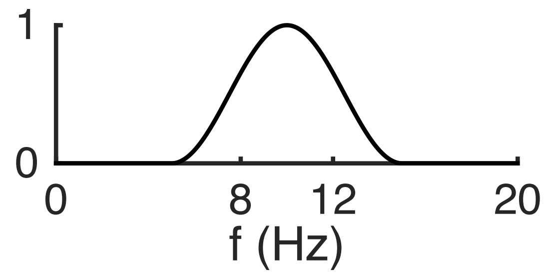
"Ideal" Filter



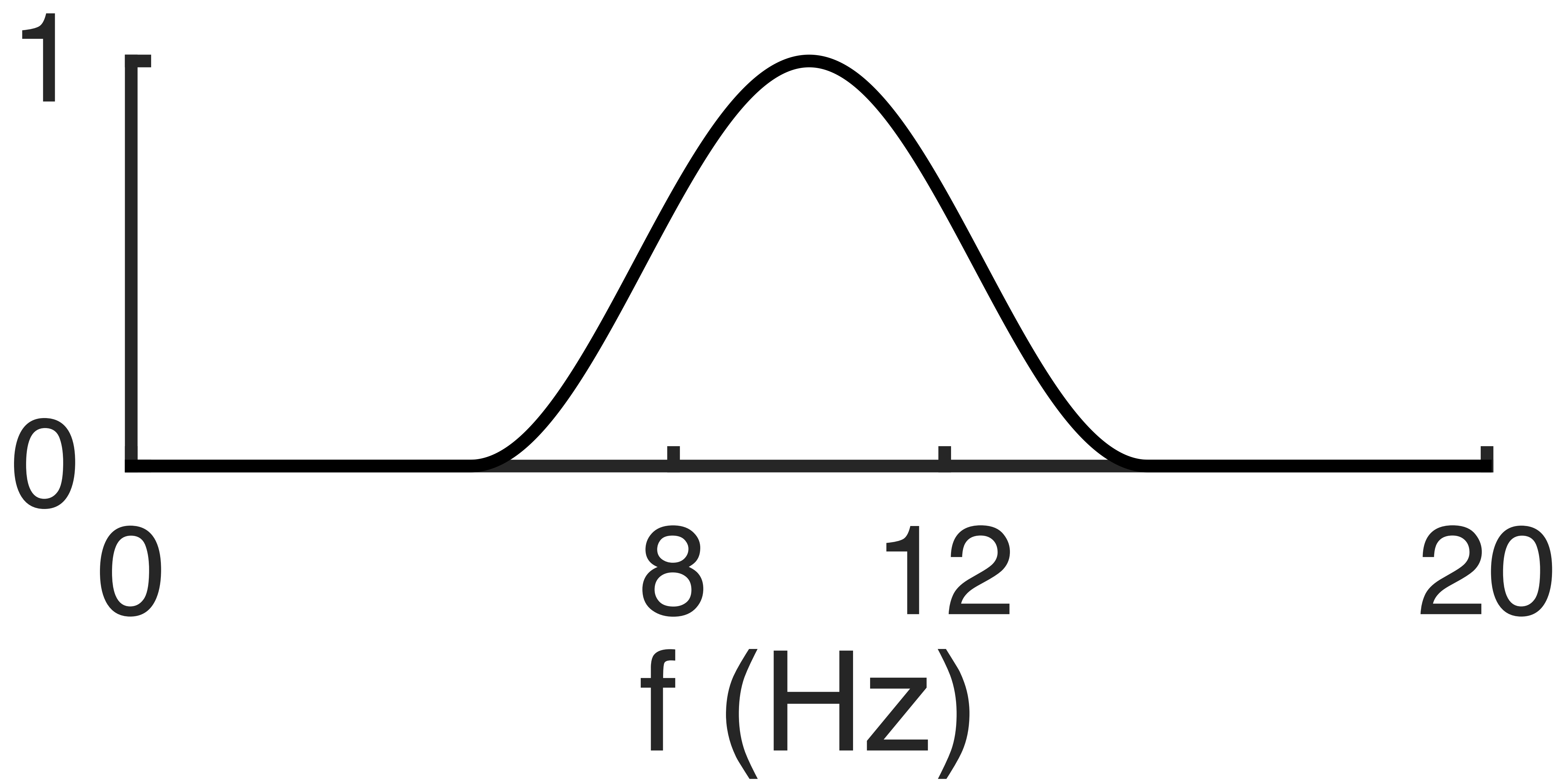
Sharp Transition



Soft Transition



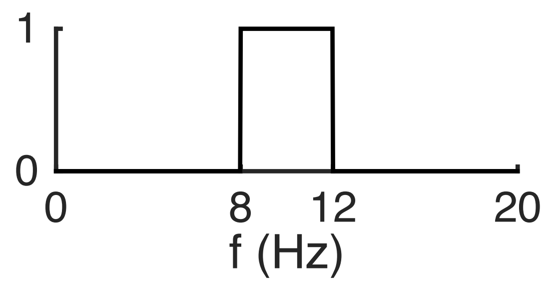
Soft Transition



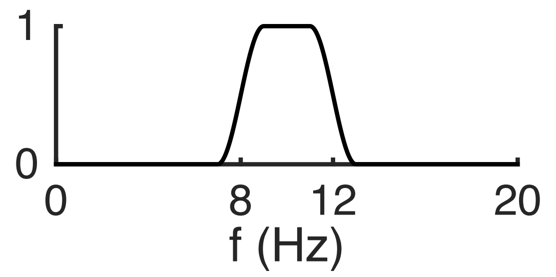
Filters: How Selective?

- *How sharp a transition?*

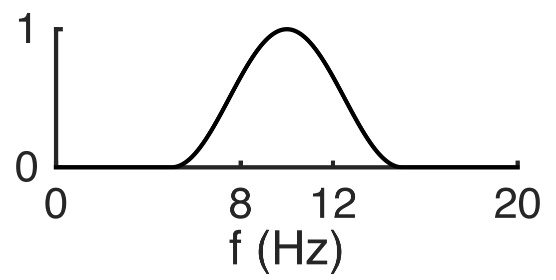
"Ideal" Filter



Sharp Transition



Soft Transition



Filters: How Do They Work?

Output of Filter:

Examples:

- Linear Combination of *Input Signal* and ***Earlier Versions*** *of the Input Signal*

$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$

- Linear Combination of *Input Signal* and ***Earlier Versions*** of the ***Output*** *Signal*

$$y[t] = \frac{1}{10}x[t] - \frac{9}{10}y[t-\Delta t]$$

- Linear Combination of *Input Signal* and ***Earlier Versions*** of both the ***Input and Output*** *Signals*

$$y[t] = x[t] - x[t-\Delta t] + x[t-2\Delta t] + \frac{99}{100}y[t-\Delta t] - \left(\frac{99}{100}\right)^2 y[t-2\Delta t]$$

Filters: How Do They Work?

Output of Filter:

Examples:

- Linear Combination of *Input Signal* and ***Earlier Versions*** *of the Input Signal*

$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$

- Linear Combination of *Input Signal* and ***Earlier Versions*** of the ***Output*** *Signal*

$$y[t] = \frac{1}{10}x[t] - \frac{9}{10}y[t-\Delta t]$$

- Linear Combination of *Input Signal* and ***Earlier Versions*** of both the ***Input and Output*** *Signals*

$$y[t] = x[t] - x[t-\Delta t] + x[t-2\Delta t] + \frac{99}{100}y[t-\Delta t] - \left(\frac{99}{100}\right)^2 y[t-2\Delta t]$$

Filters: How Do They Work?

Output of Filter:

- Linear Combination of *Input Signal* and ***Earlier Versions*** *of the Input Signal*
- Linear Combination of *Input Signal* and ***Earlier Versions*** of the ***Output*** *Signal*
- Linear Combination of *Input Signal* and ***Earlier Versions*** of both the ***Input and Output*** *Signals*

Examples:

$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t - \Delta t]$$

$$y[t] = \frac{1}{10}x[t] - \frac{9}{10}y[t - \Delta t]$$

$$y[t] = x[t] - x[t - \Delta t] + x[t - 2\Delta t] + \frac{99}{100}y[t - \Delta t] - \left(\frac{99}{100}\right)^2 y[t - 2\Delta t]$$

Filters: How Do They Work?

Output of Filter:

Examples:

- Linear Combination of *Input Signal* and ***Earlier Versions*** *of the Input Signal*

$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$

- Linear Combination of *Input Signal* and ***Earlier Versions*** of the ***Output*** *Signal*

$$y[t] = \frac{1}{10}x[t] - \frac{9}{10}y[t-\Delta t]$$

- Linear Combination of *Input Signal* and ***Earlier Versions*** of both the ***Input and Output*** *Signals*

$$y[t] = x[t] - x[t-\Delta t] + x[t-2\Delta t] + \frac{99}{100}y[t-\Delta t] - \left(\frac{99}{100}\right)^2 y[t-2\Delta t]$$

Filters: How Do They Work?

Output of Filter:

- Linear Combination of *Input Signal* and ***Earlier Versions*** *of the Input Signal*
- Linear Combination of *Input Signal* and ***Earlier Versions*** of the ***Output*** *Signal*
- Linear Combination of *Input Signal* and ***Earlier Versions*** of both the ***Input and Output*** *Signals*

Examples:

$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$

$$y[t] = \frac{1}{10}x[t] - \frac{9}{10}y[t-\Delta t]$$

$$y[t] = x[t] - x[t-\Delta t] + x[t-2\Delta t] + \frac{99}{100}y[t-\Delta t] - \left(\frac{99}{100}\right)^2 y[t-2\Delta t]$$

Filters: How Do They Work?

Output of Filter:

- Linear Combination of *Input Signal* and ***Earlier Versions*** *of the Input Signal*
- Linear Combination of *Input Signal* and ***Earlier Versions*** of the ***Output*** *Signal*
- Linear Combination of *Input Signal* and ***Earlier Versions*** of both the ***Input and Output*** *Signals*

Examples:

$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$

$$y[t] = \frac{1}{10}x[t] - \frac{9}{10}y[t-\Delta t]$$

$$y[t] = x[t] - x[t-\Delta t] + x[t-2\Delta t] + \frac{99}{100}y[t-\Delta t] - \left(\frac{99}{100}\right)^2 y[t-2\Delta t]$$

Filters: How Do They Work?

Output of Filter:

- Linear Combination of *Input Signal* and ***Earlier Versions*** *of the Input Signal*
- Linear Combination of *Input Signal* and ***Earlier Versions*** of the ***Output*** *Signal*
- Linear Combination of *Input Signal* and ***Earlier Versions*** of both the ***Input and Output*** *Signals*

Examples:

$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$

$$y[t] = \frac{1}{10}x[t] - \frac{9}{10}y[t-\Delta t]$$

$$y[t] = x[t] - x[t-\Delta t] + x[t-2\Delta t] + \frac{99}{100}y[t-\Delta t] - \left(\frac{99}{100}\right)^2 y[t-2\Delta t]$$

Filters: How Do They Work?

Output of Filter:

- Linear Combination of *Input Signal* and ***Earlier Versions*** *of the Input Signal*
- Linear Combination of *Input Signal* and ***Earlier Versions*** of the ***Output*** *Signal*
- Linear Combination of *Input Signal* and ***Earlier Versions*** of both the ***Input and Output*** *Signals*

Examples:

$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$

$$y[t] = \frac{1}{10}x[t] - \frac{9}{10}y[t-\Delta t]$$

$$y[t] = x[t] - x[t-\Delta t] + x[t-2\Delta t] + \frac{99}{100}y[t-\Delta t] - \left(\frac{99}{100}\right)^2 y[t-2\Delta t]$$

Filters: How Do They Work?

Output of Filter:

- Linear Combination of *Input Signal* and ***Earlier Versions*** *of the Input Signal*
- Linear Combination of *Input Signal* and ***Earlier Versions*** of the ***Output*** *Signal*
- Linear Combination of *Input Signal* and ***Earlier Versions*** of both the ***Input and Output*** *Signals*

Examples:

$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$

$$y[t] = \frac{1}{10}x[t] - \frac{9}{10}y[t-\Delta t]$$

$$y[t] = x[t] - x[t-\Delta t] + x[t-2\Delta t] + \frac{99}{100}y[t-\Delta t] - \left(\frac{99}{100}\right)^2 y[t-2\Delta t]$$

Filters: How Do They Work?

Output of Filter:

- Linear Combination of *Input Signal* and ***Earlier Versions*** *of the Input Signal*
- Linear Combination of *Input Signal* and ***Earlier Versions*** of the ***Output*** *Signal*
- Linear Combination of *Input Signal* and ***Earlier Versions*** of both the ***Input and Output*** *Signals*

Examples:

$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t - \Delta t]$$

$$y[t] = \frac{1}{10}x[t] - \frac{9}{10}y[t - \Delta t]$$

$$y[t] = x[t] - x[t - \Delta t] + x[t - 2\Delta t] + \frac{99}{100}y[t - \Delta t] - \left(\frac{99}{100}\right)^2 y[t - 2\Delta t]$$

Example: Two-Point Moving Average

$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t - \Delta t]$$

What to Expect:

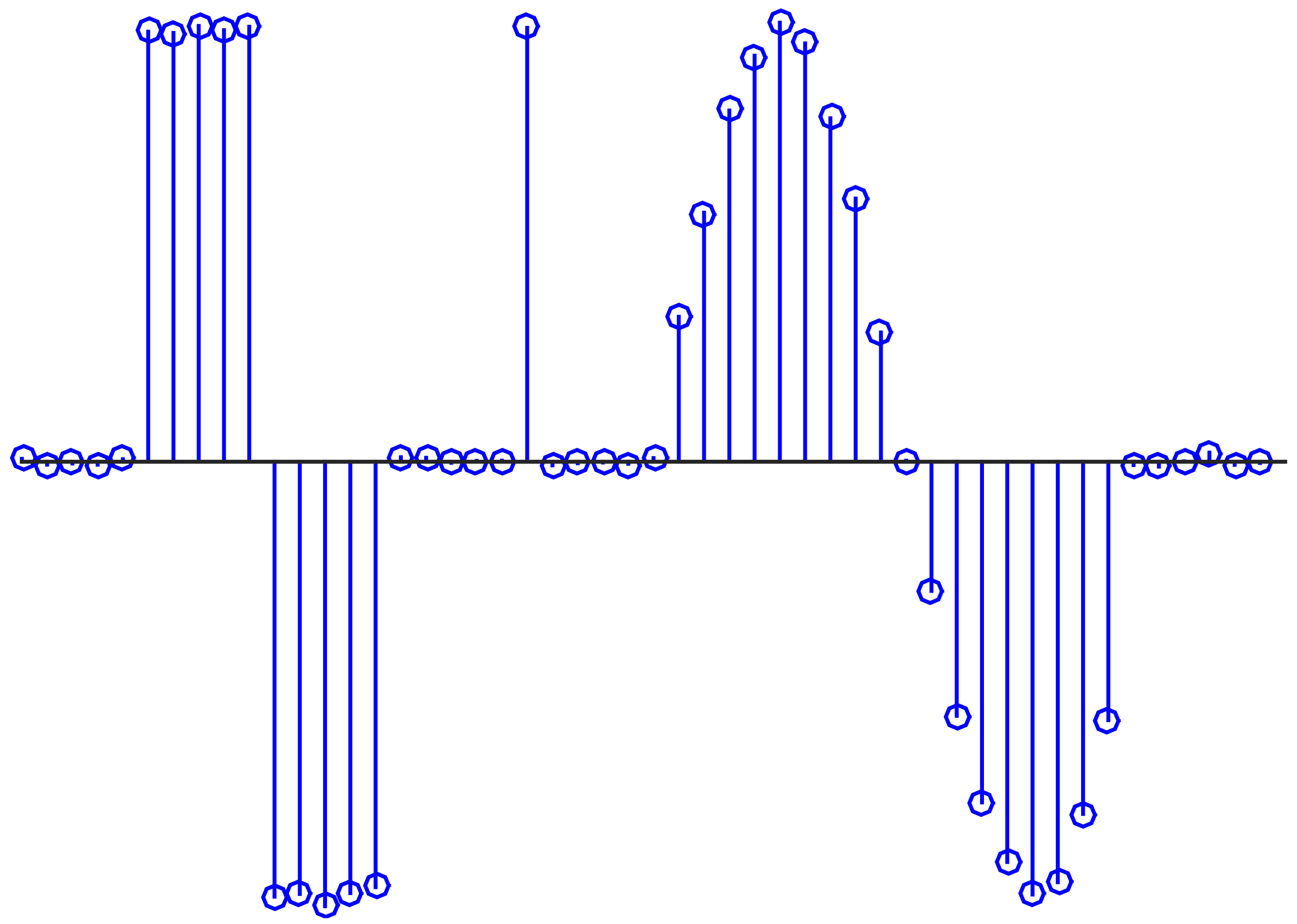
- Smooth over rough patches
- Soften sudden changes
- Leave slowly varying signals largely unchanged
- Low Pass Filter?

Example: Two-Point Moving Average

$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t - \Delta t]$$

Example: Two-Point Moving Average

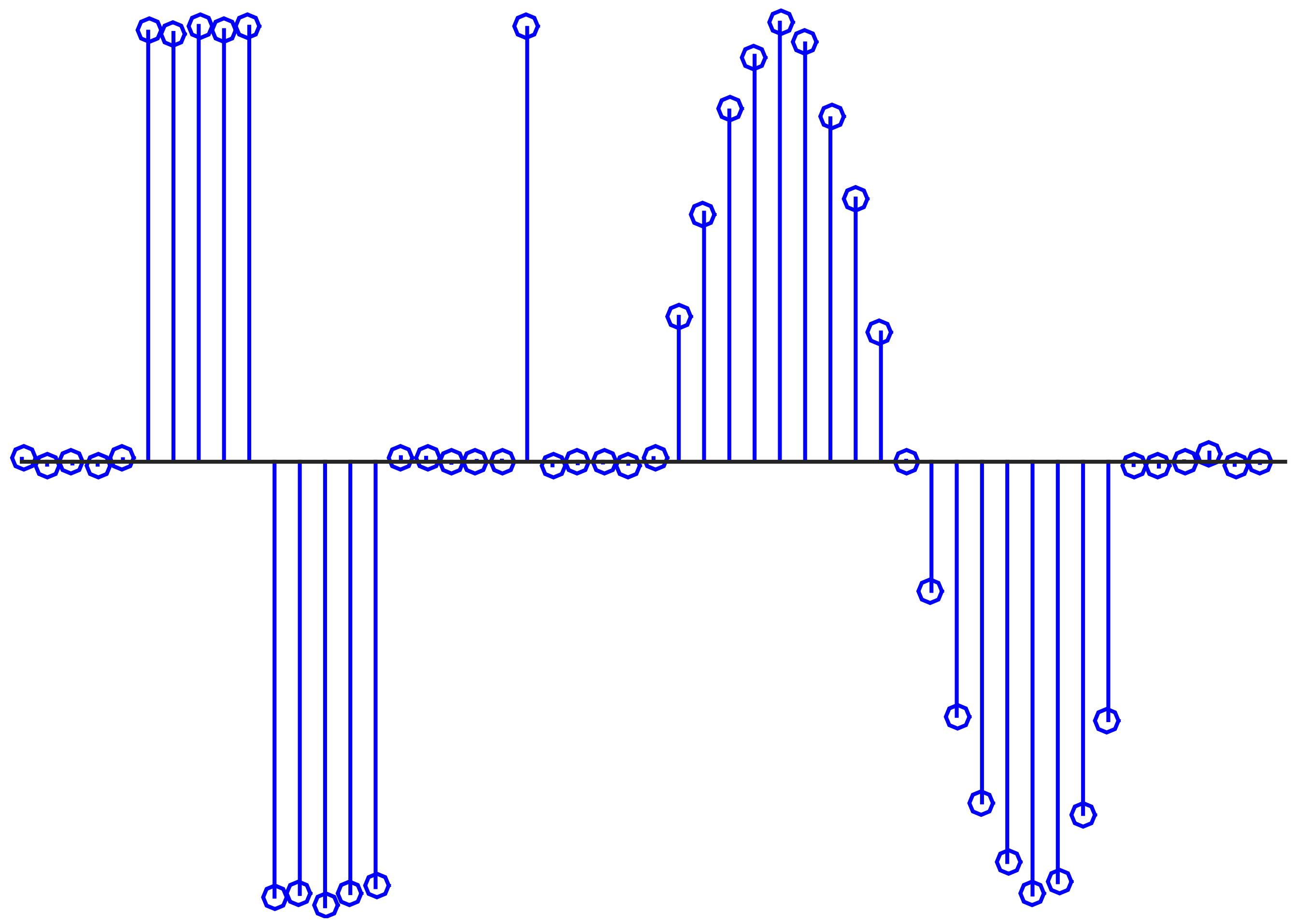
$x[t]$



$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t - \Delta t]$$

Example: Two-Point Moving Average

$x[t]$

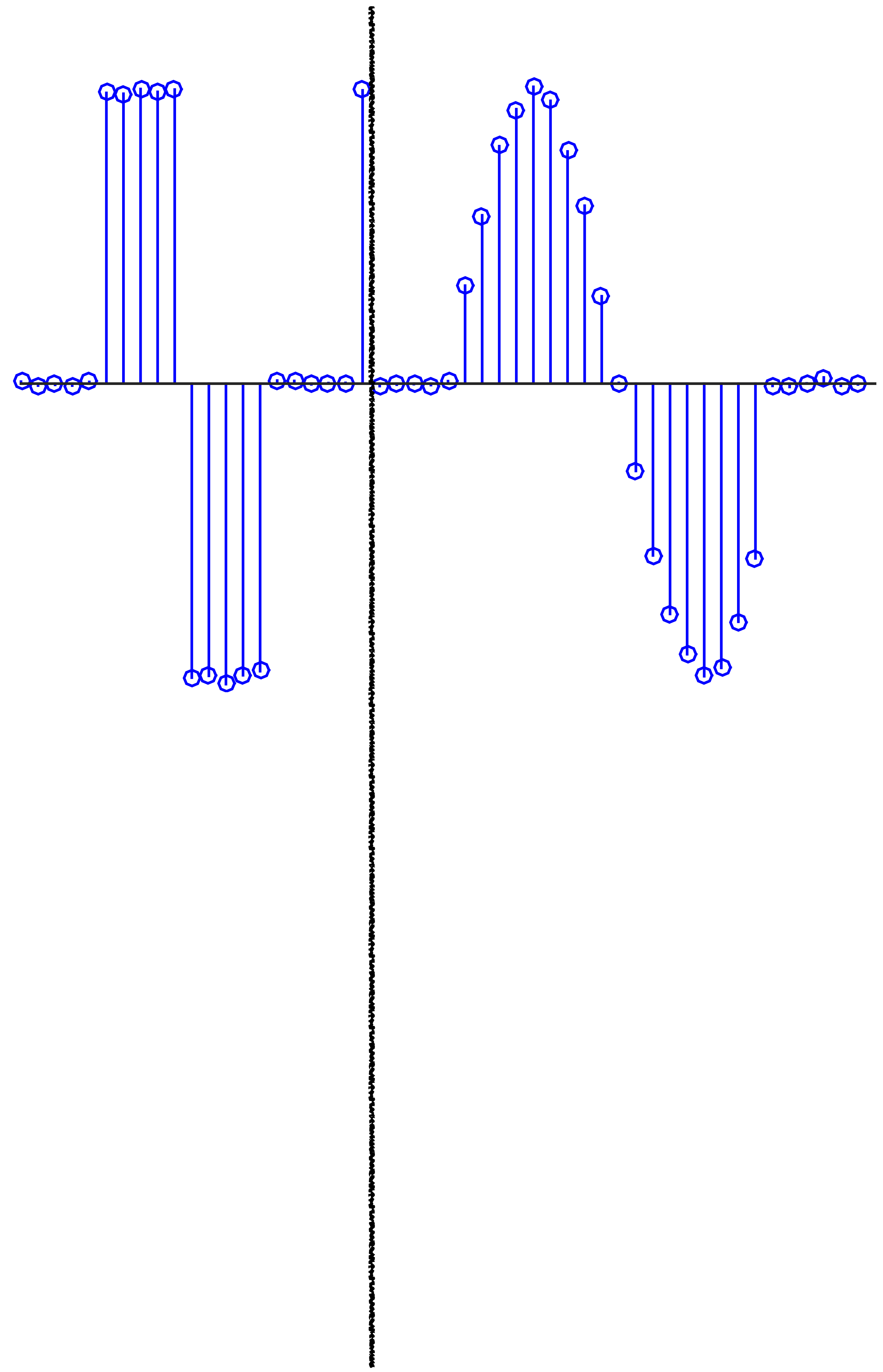


$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t - \Delta t]$$

$x[t - \Delta t]$

Example: Two-Point Moving Average

$x[t]$



$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t - \Delta t]$$

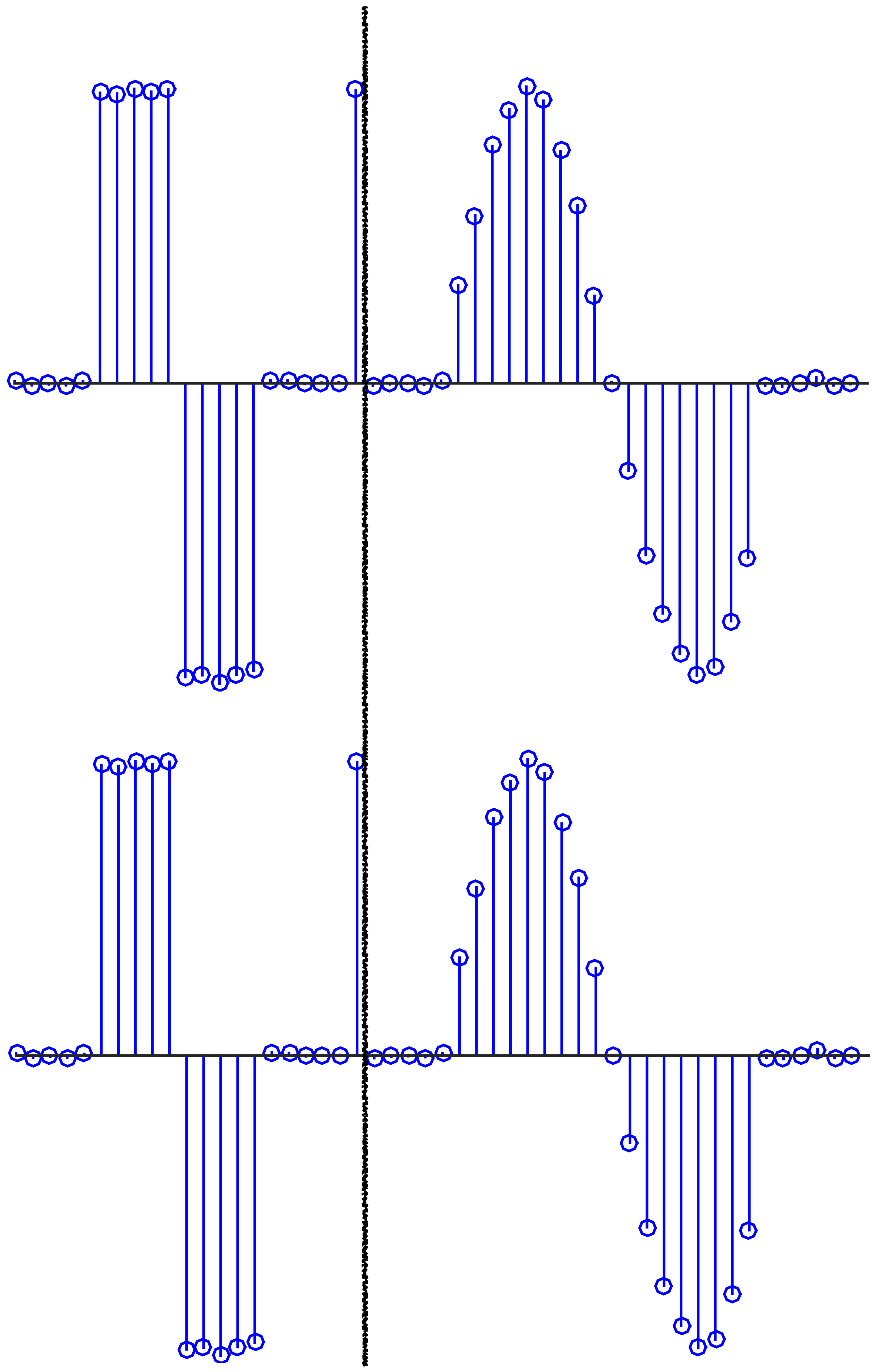
$x[t - \Delta t]$

Example: Two-Point Moving Average

$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$

$x[t]$

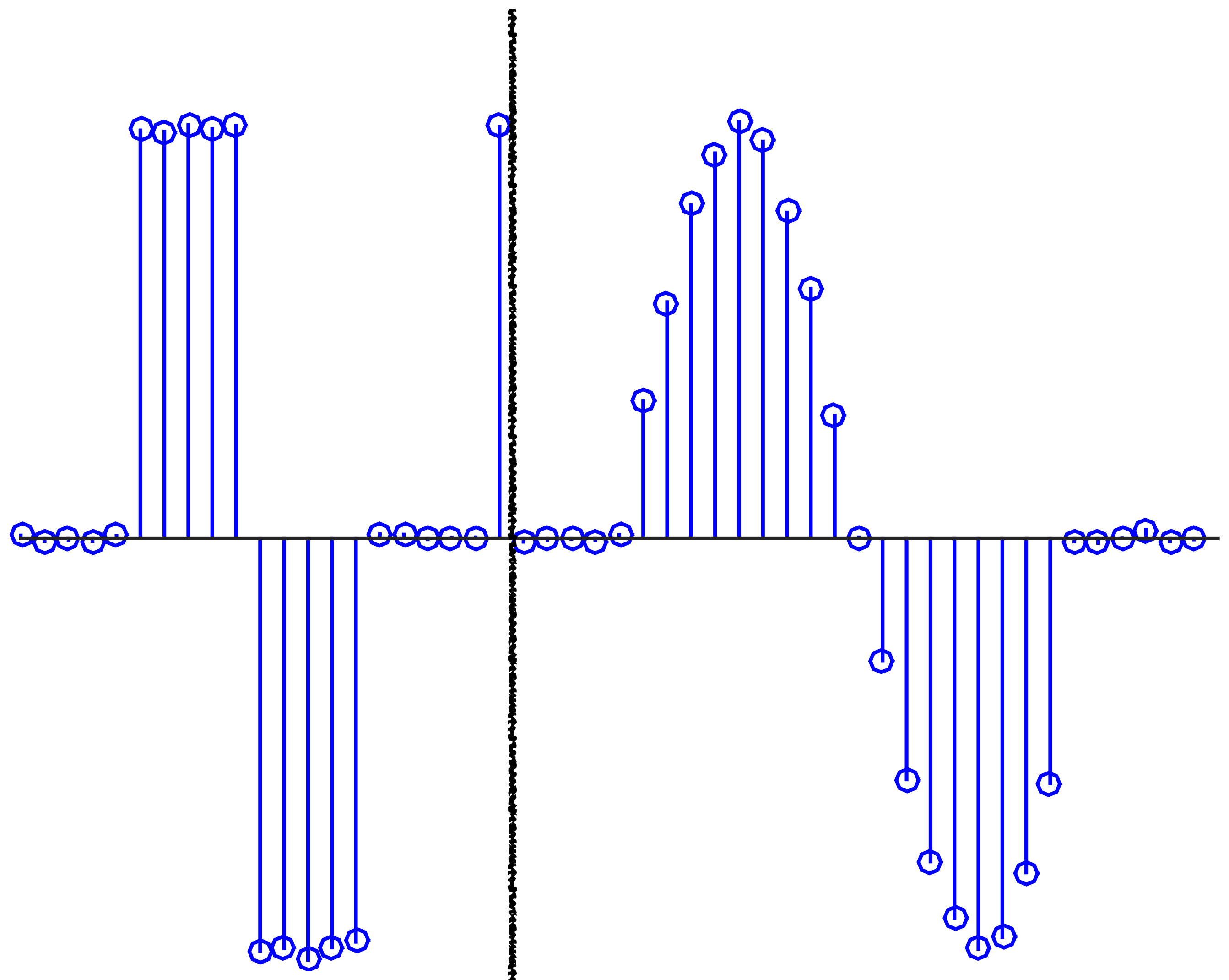
$x[t-\Delta t]$



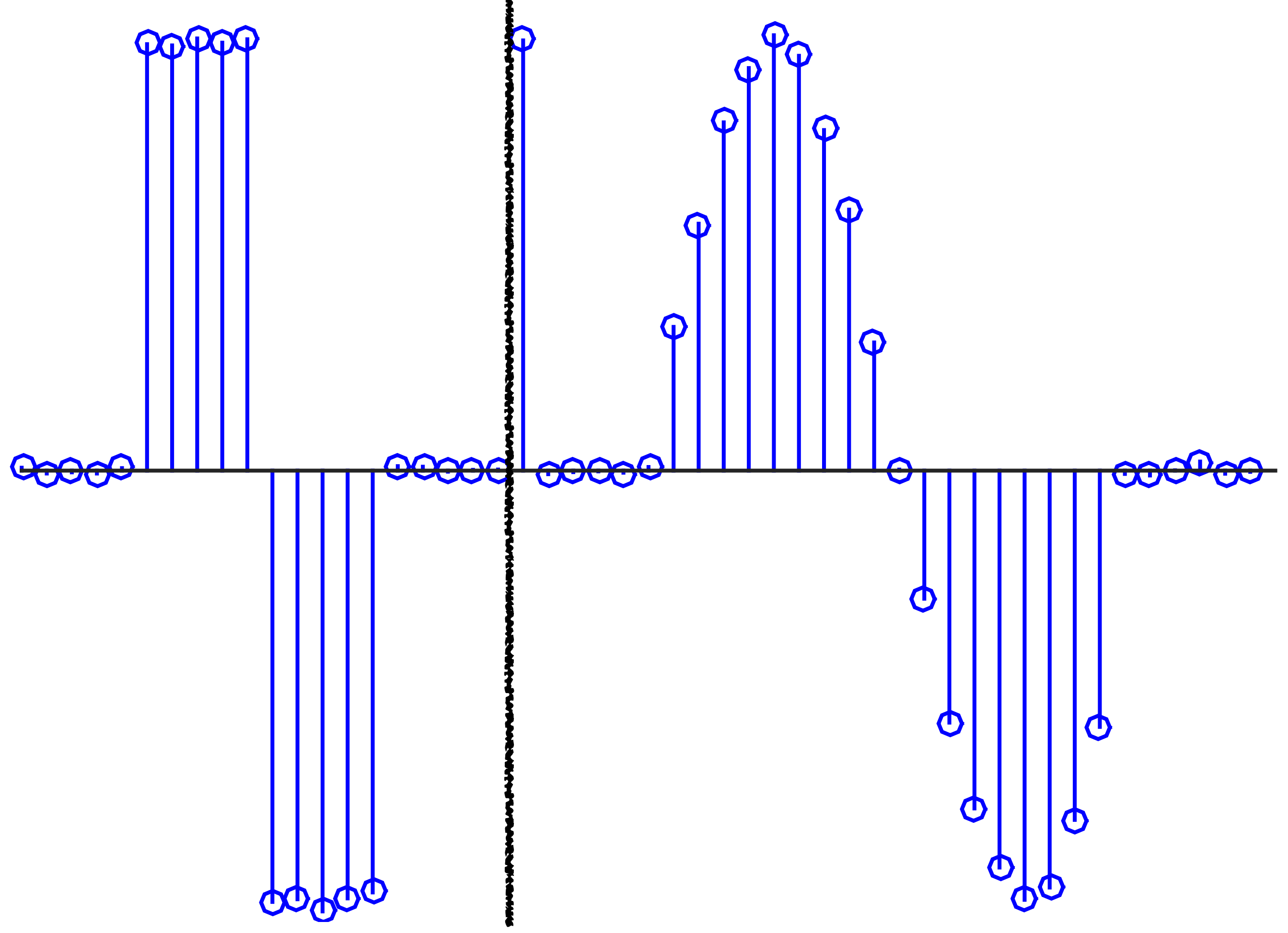
Example: Two-Point Moving Average

$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$

$x[t]$



$x[t-\Delta t]$

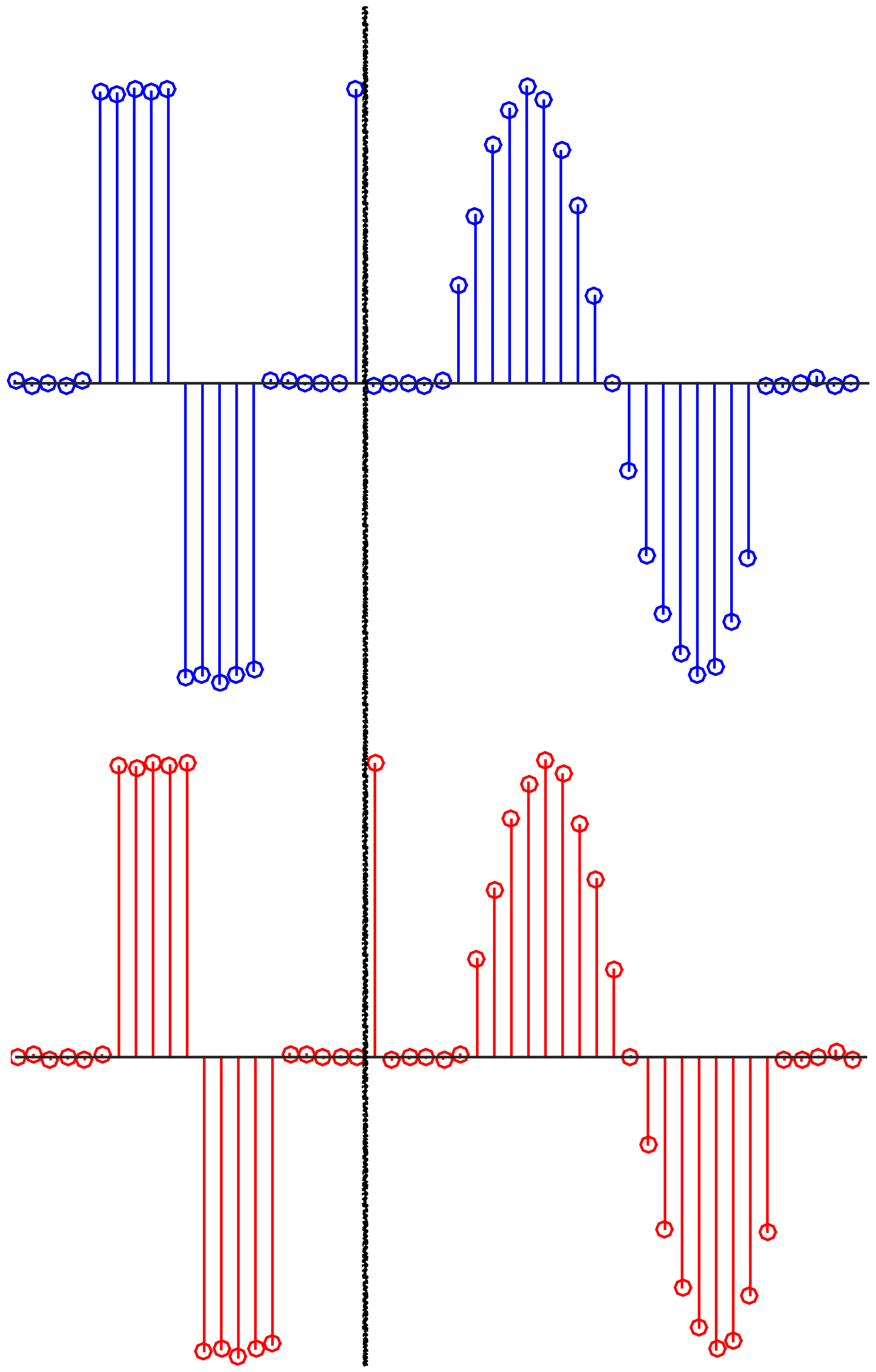


Example: Two-Point Moving Average

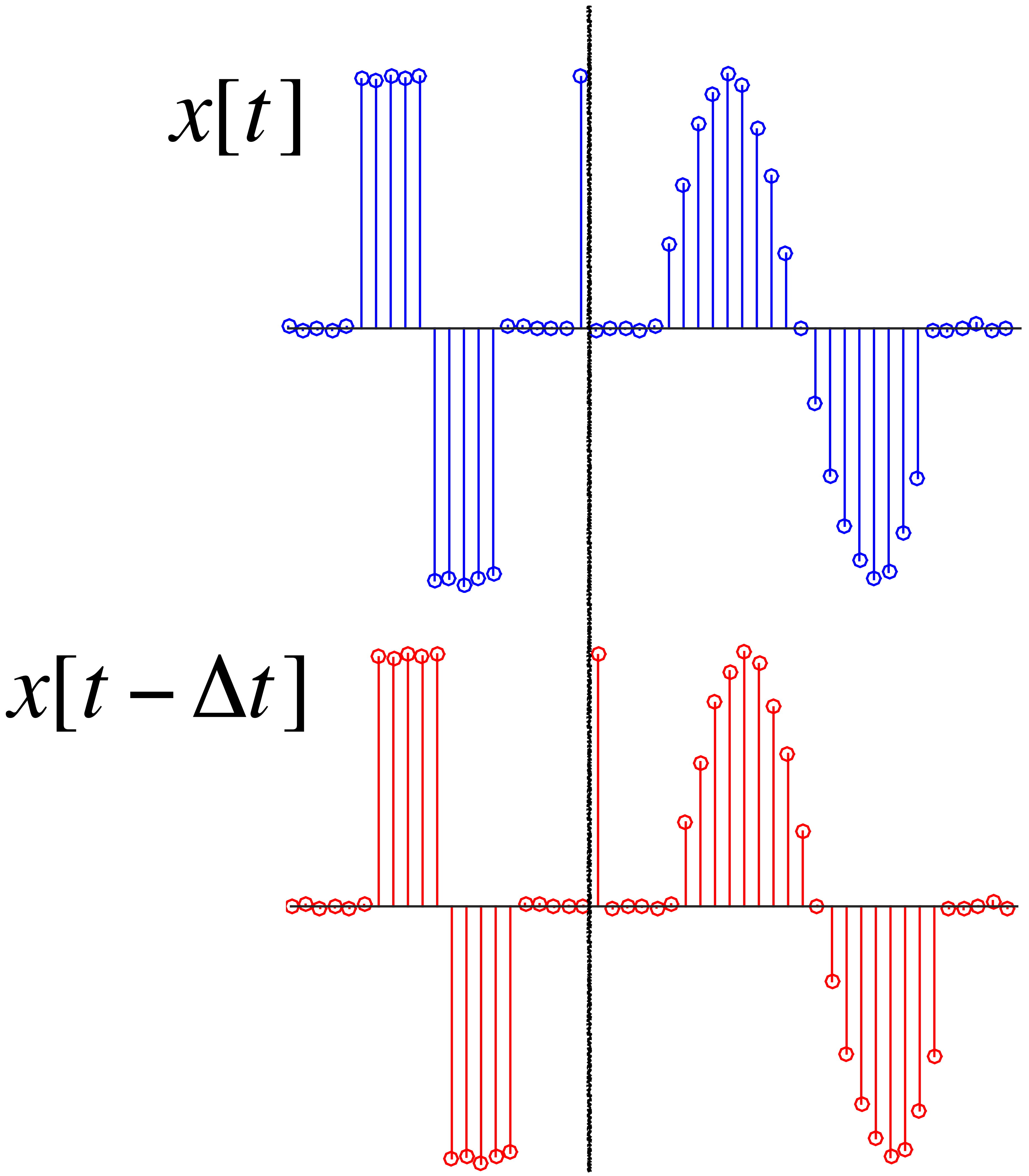
$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$

$x[t]$

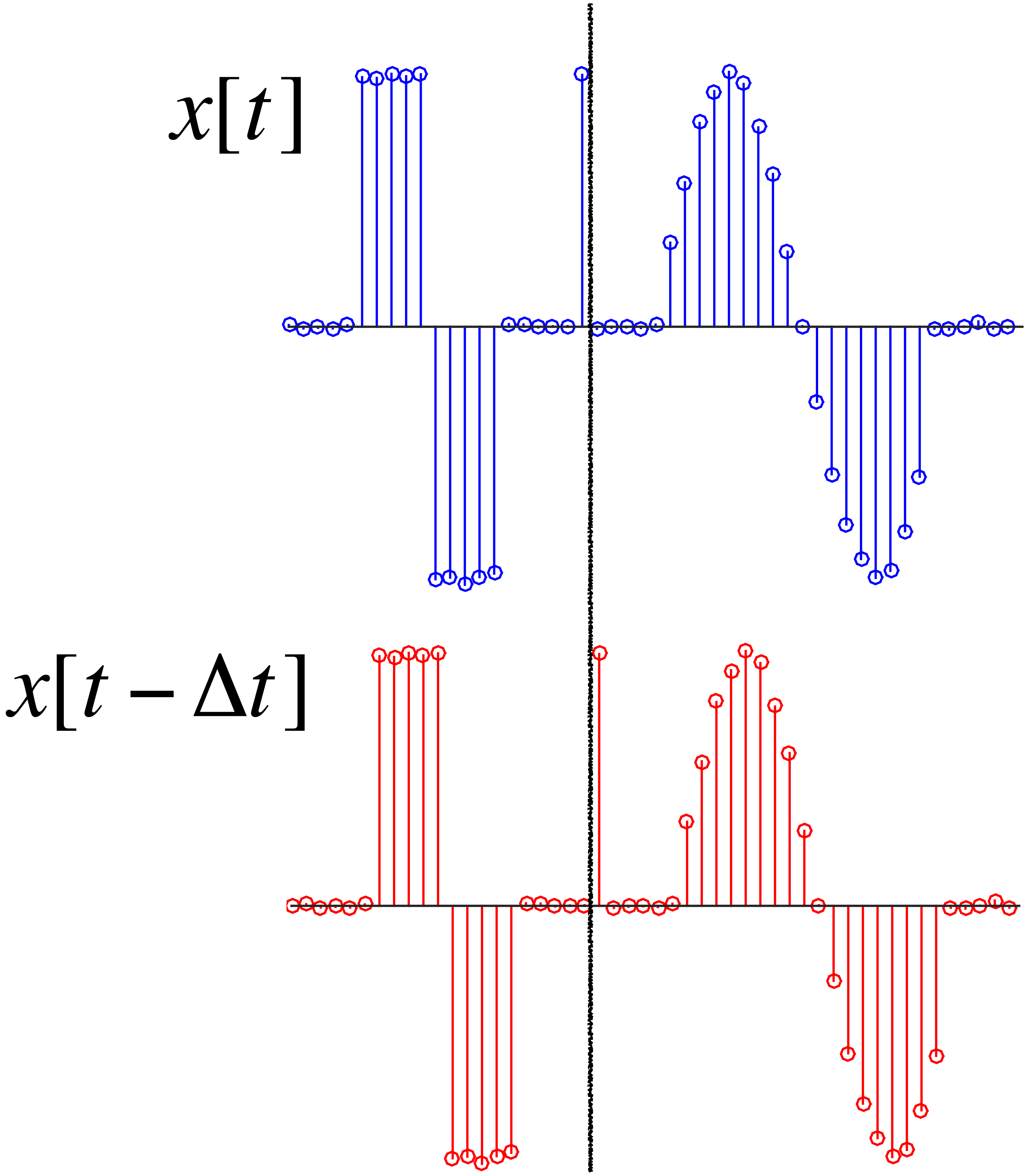
$x[t-\Delta t]$

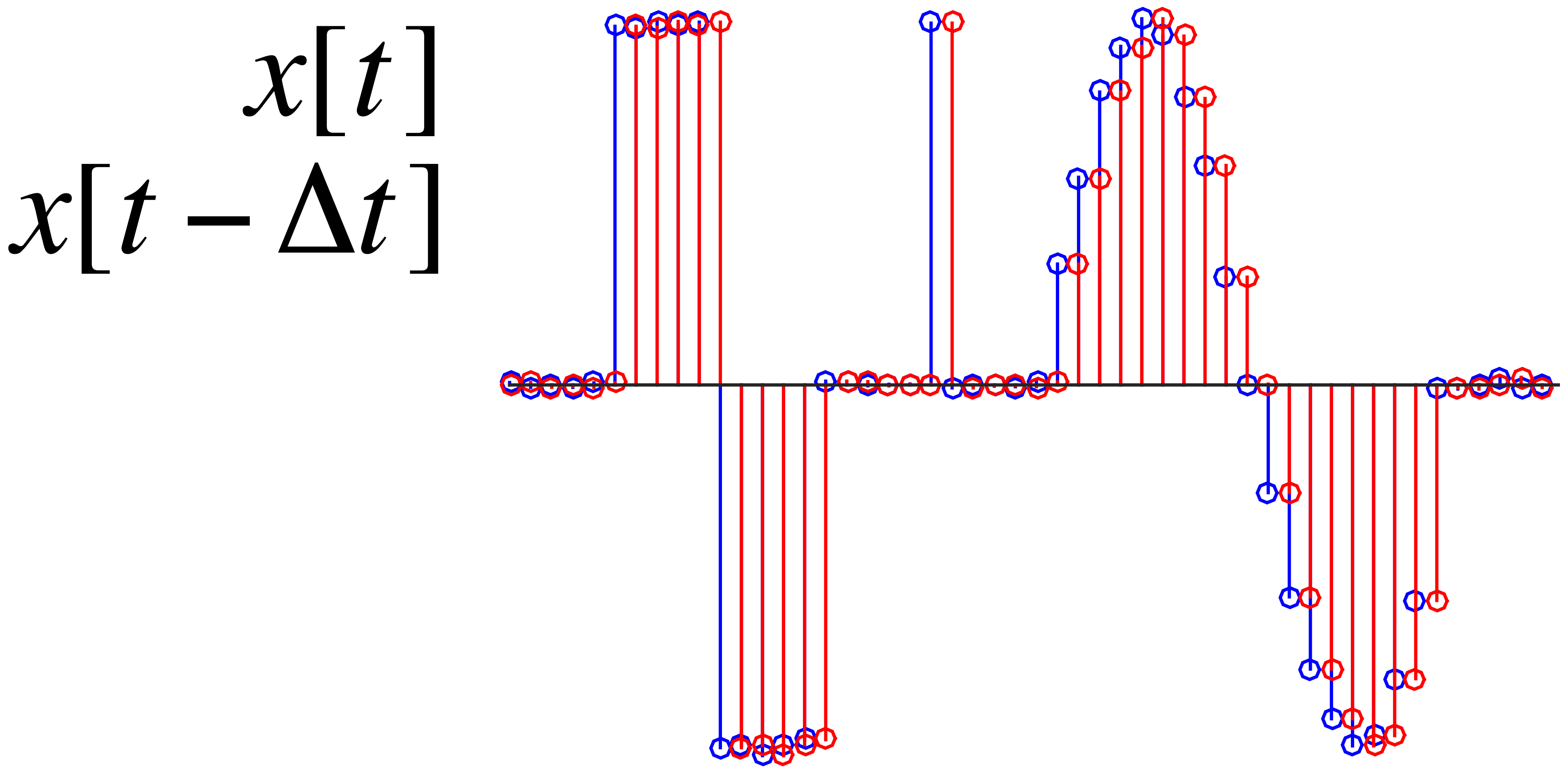


Example: Two-Point Moving Average



Example: Two-Point Moving Average

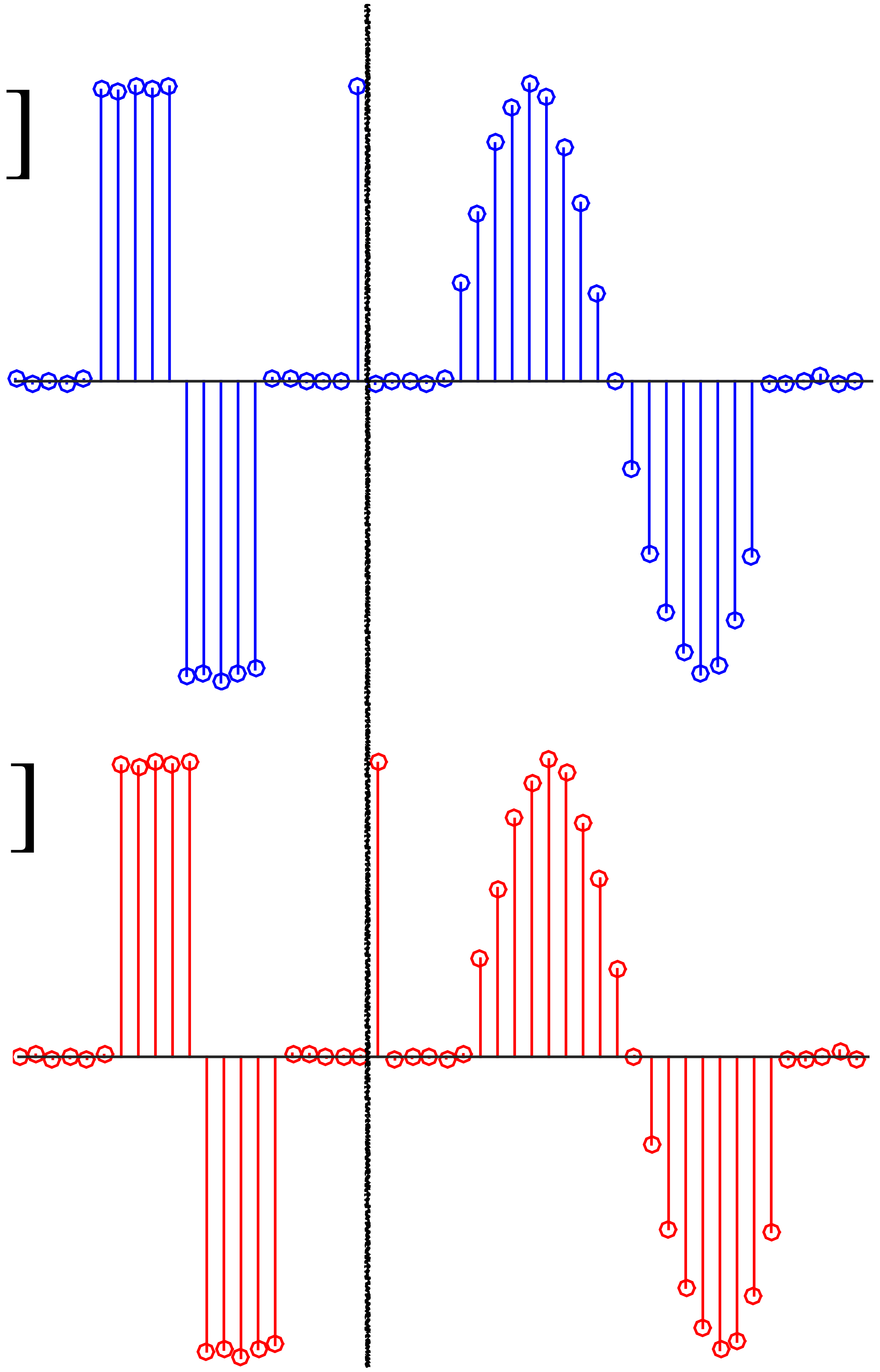




Example: Two-Point Moving Average

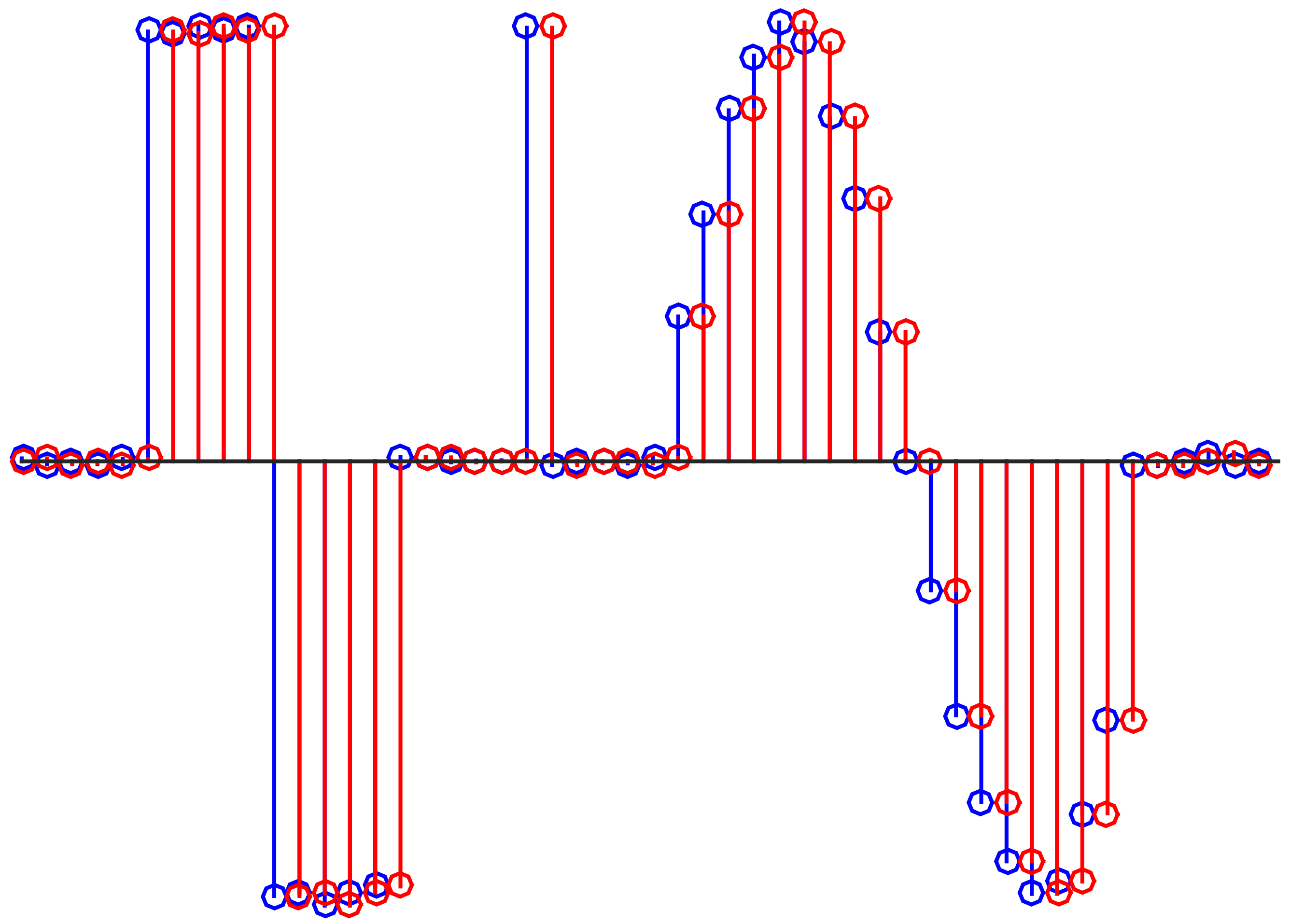
$x[t]$

$x[t - \Delta t]$



$x[t]$

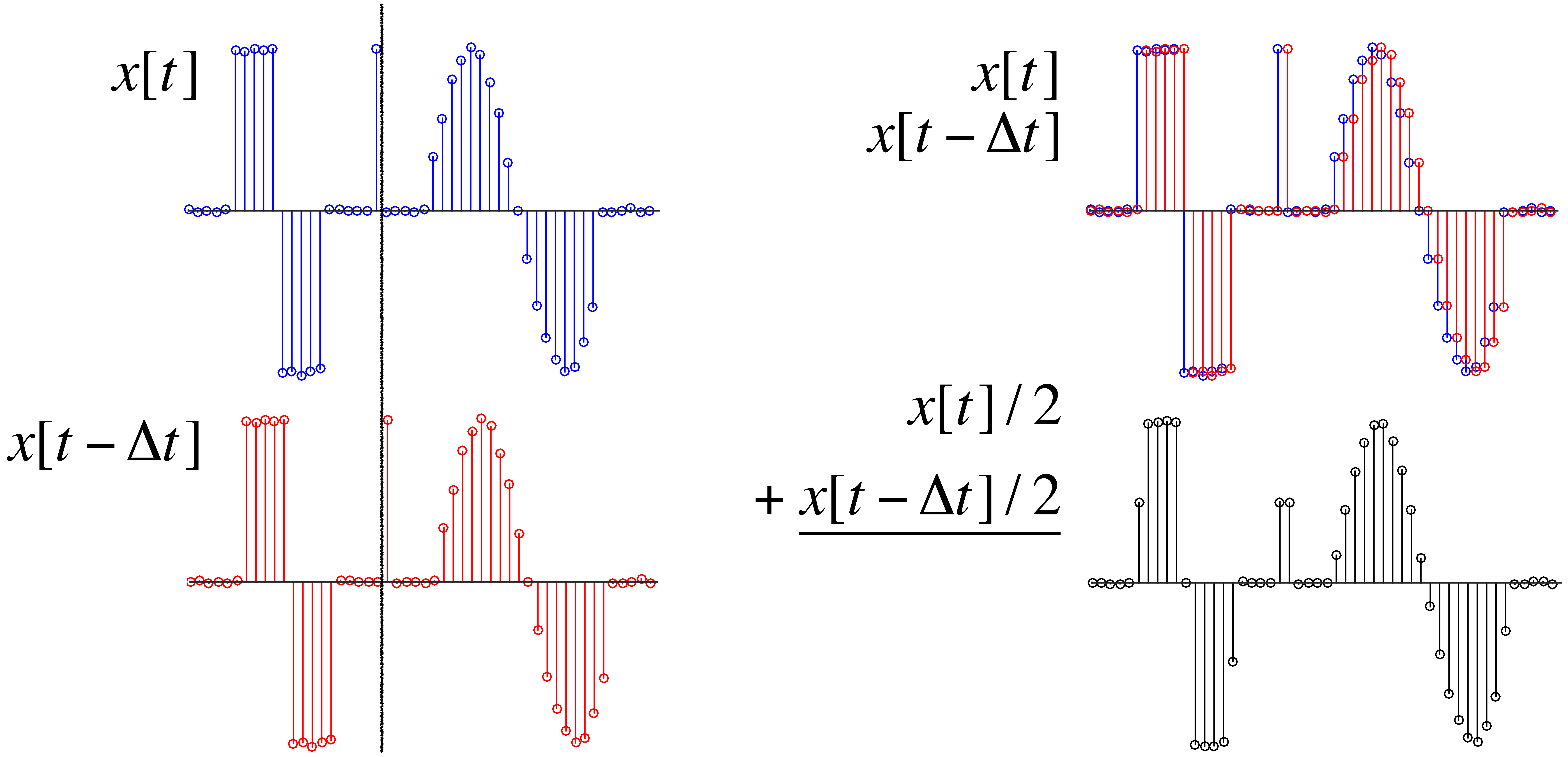
$x[t - \Delta t]$



$$x[t]/2$$

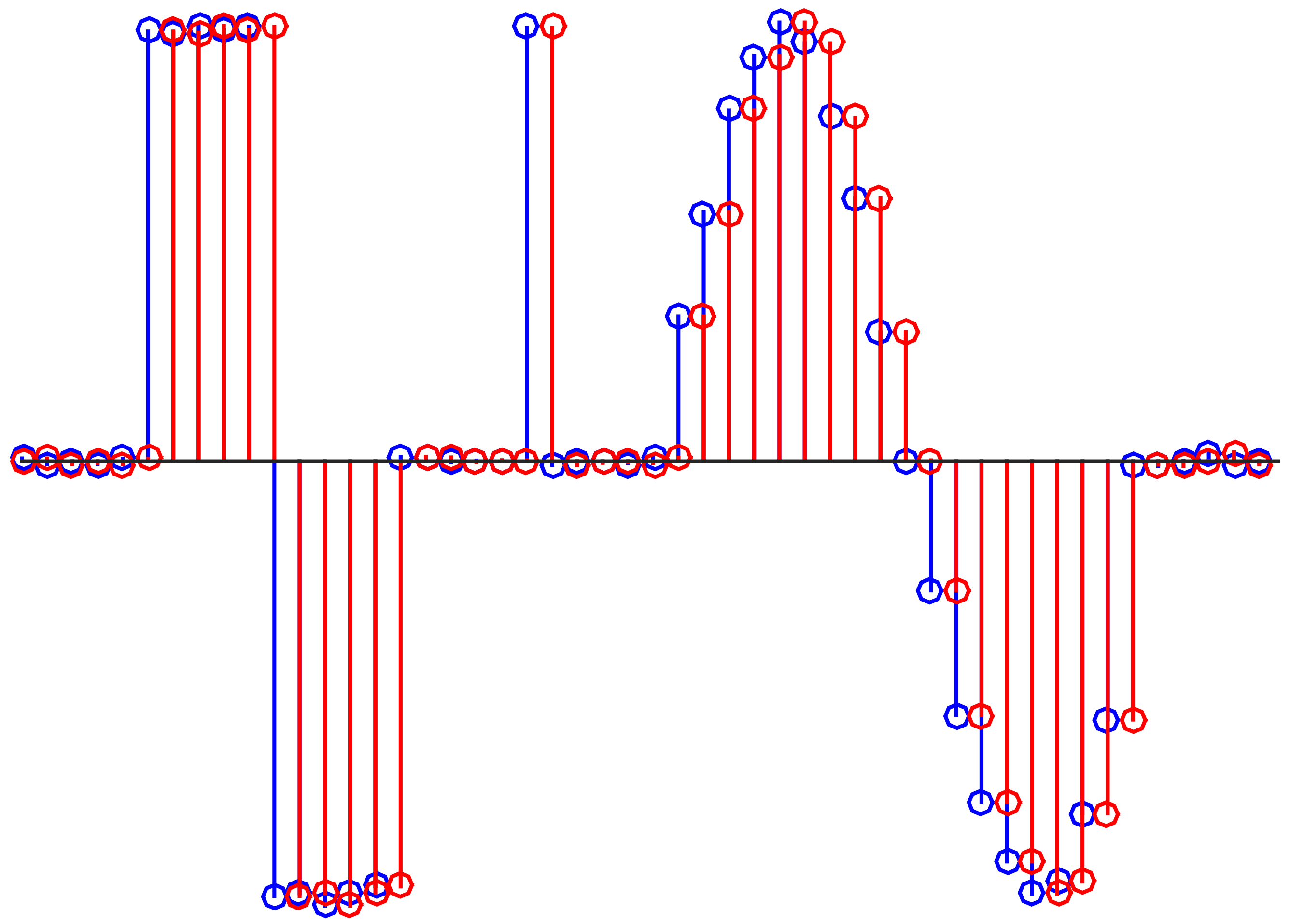
$$\underline{+\, x[t - \Delta t]/2}$$

Example: Two-Point Moving Average

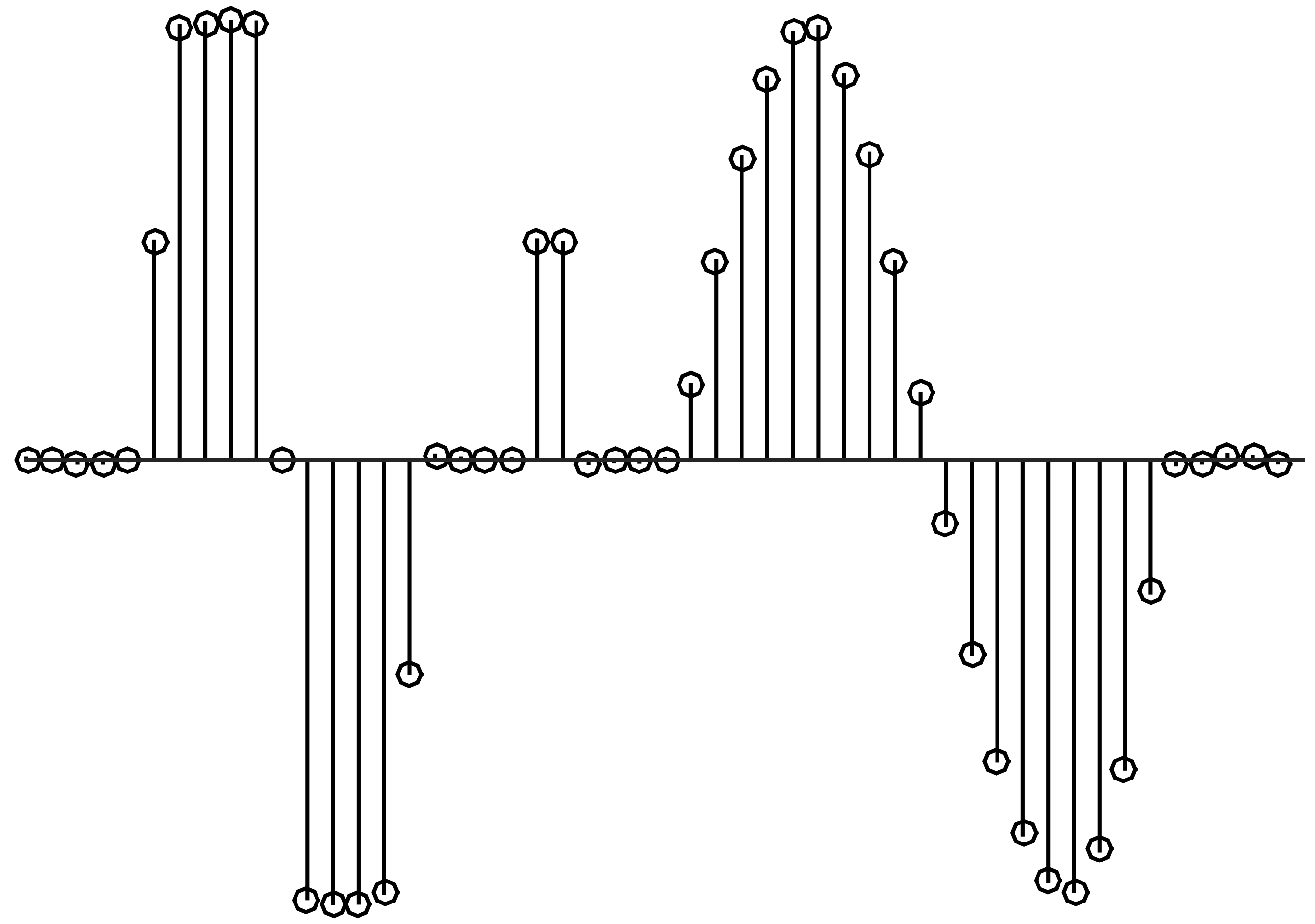


Example: Two-Point Moving Average

$x[t]$
$x[t-\Delta t]$

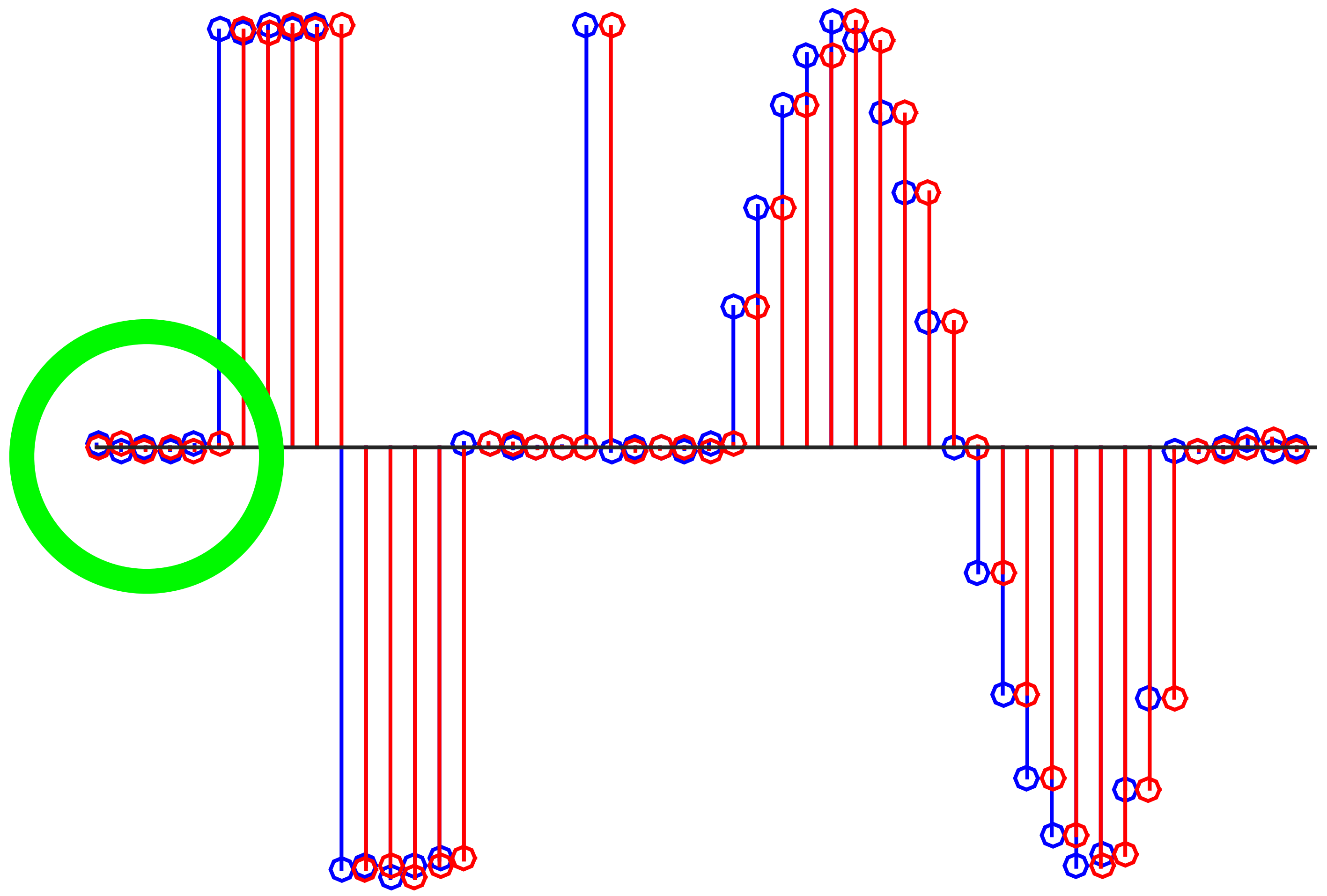


$$y[t]=\frac{1}{2}x[t]+\frac{1}{2}x[t-\Delta t]$$

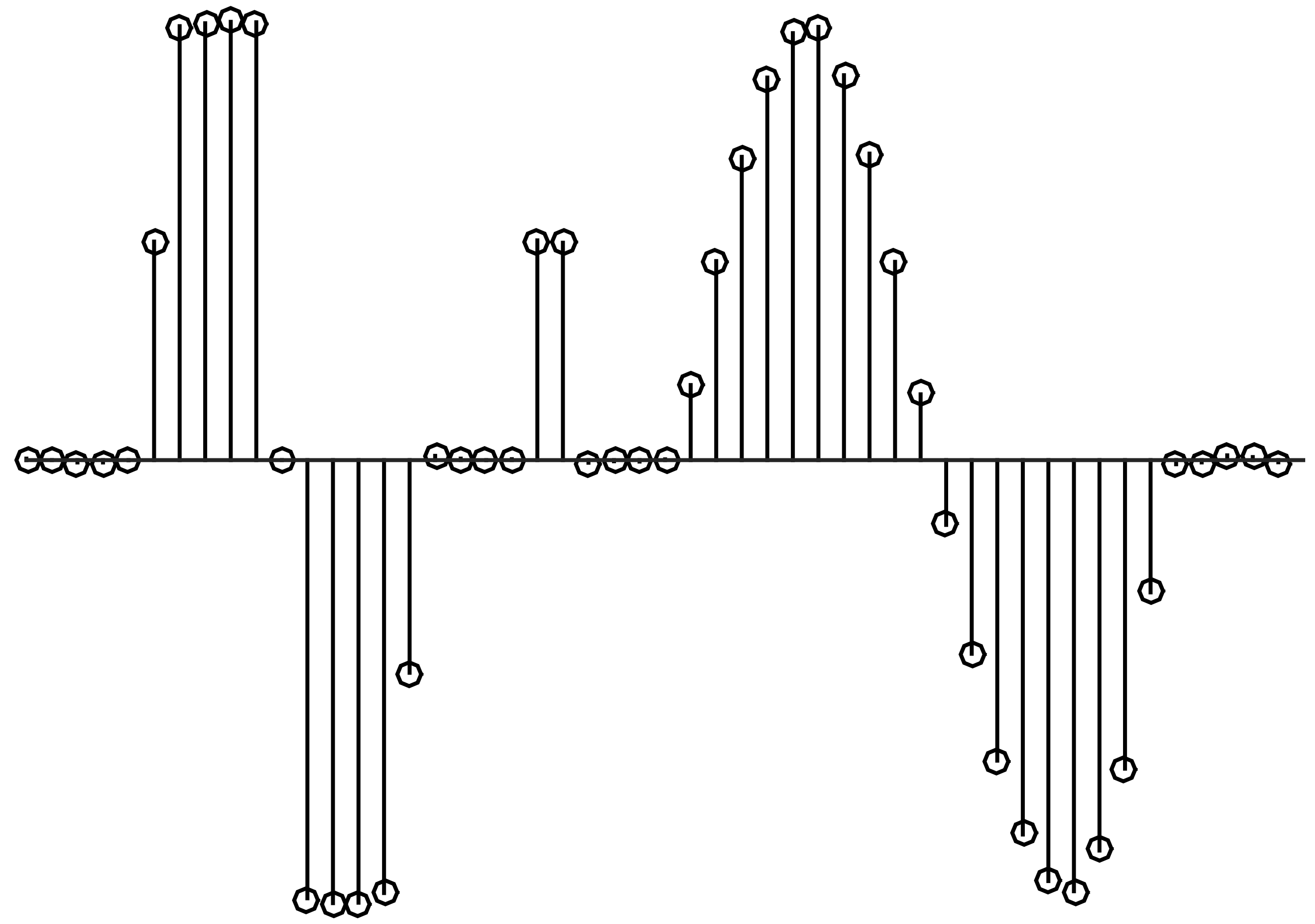


Example: Two-Point Moving Average

$x[t]$
$x[t-\Delta t]$

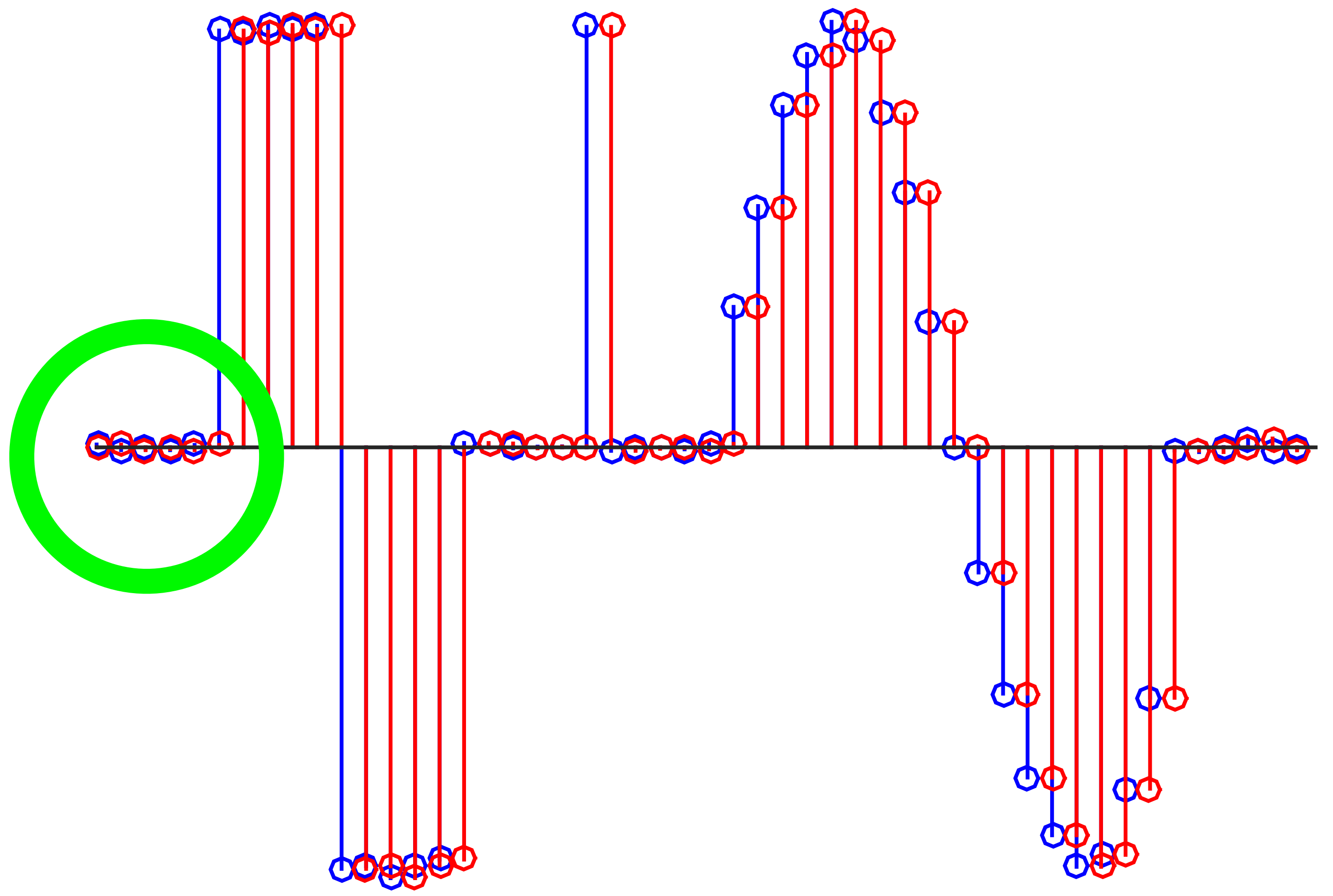


$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$

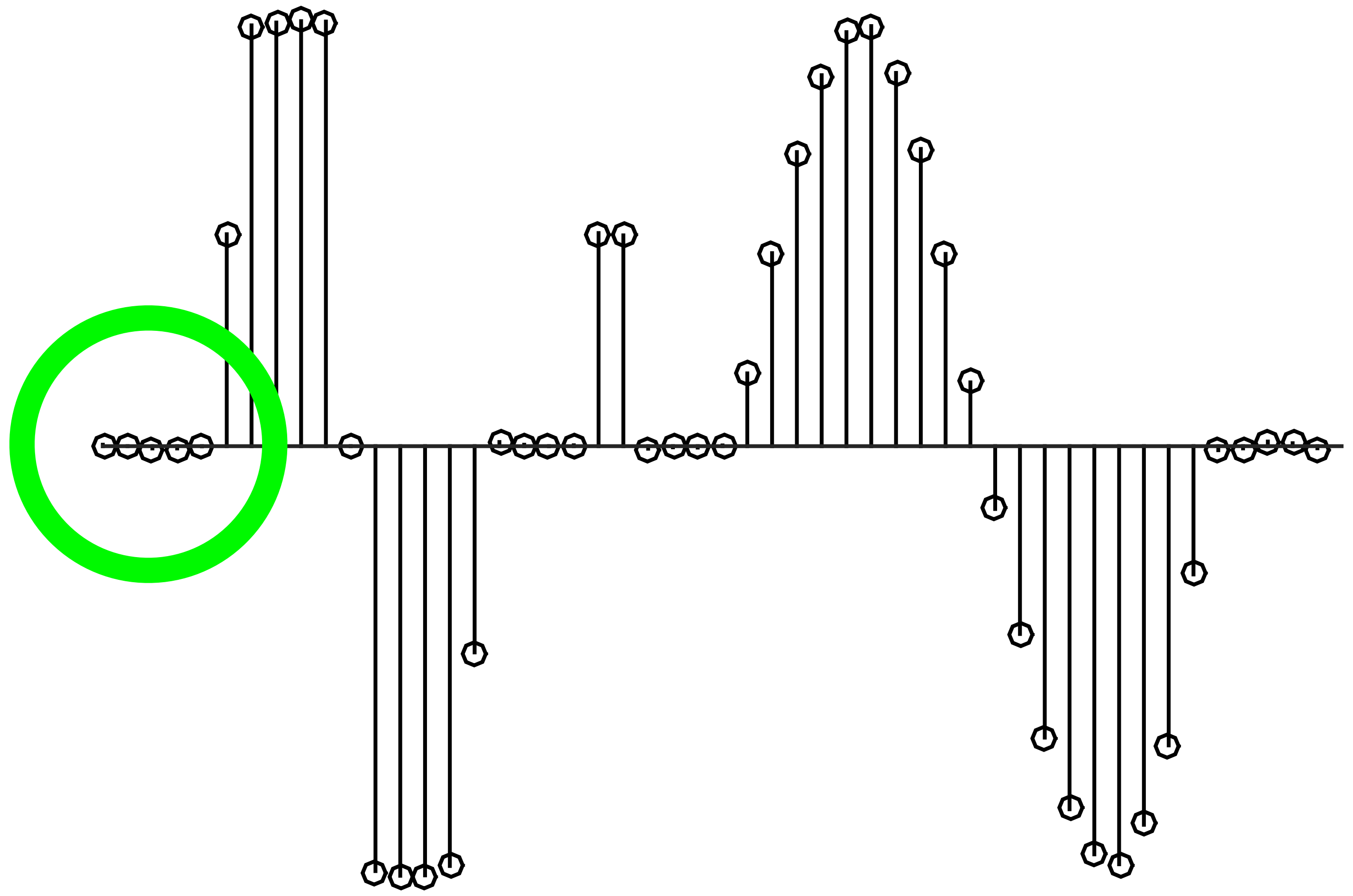


Example: Two-Point Moving Average

$x[t]$
$x[t-\Delta t]$

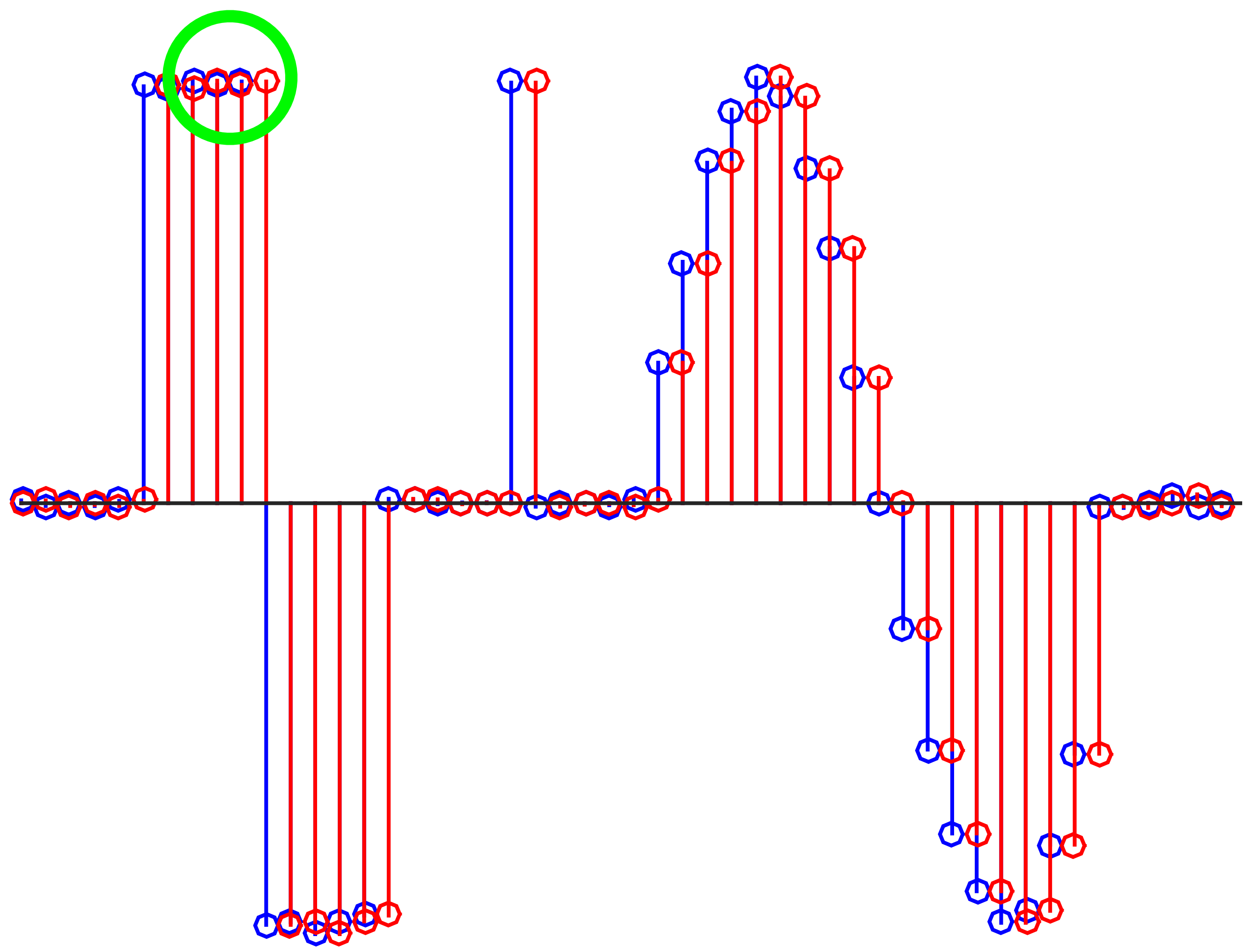


$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$

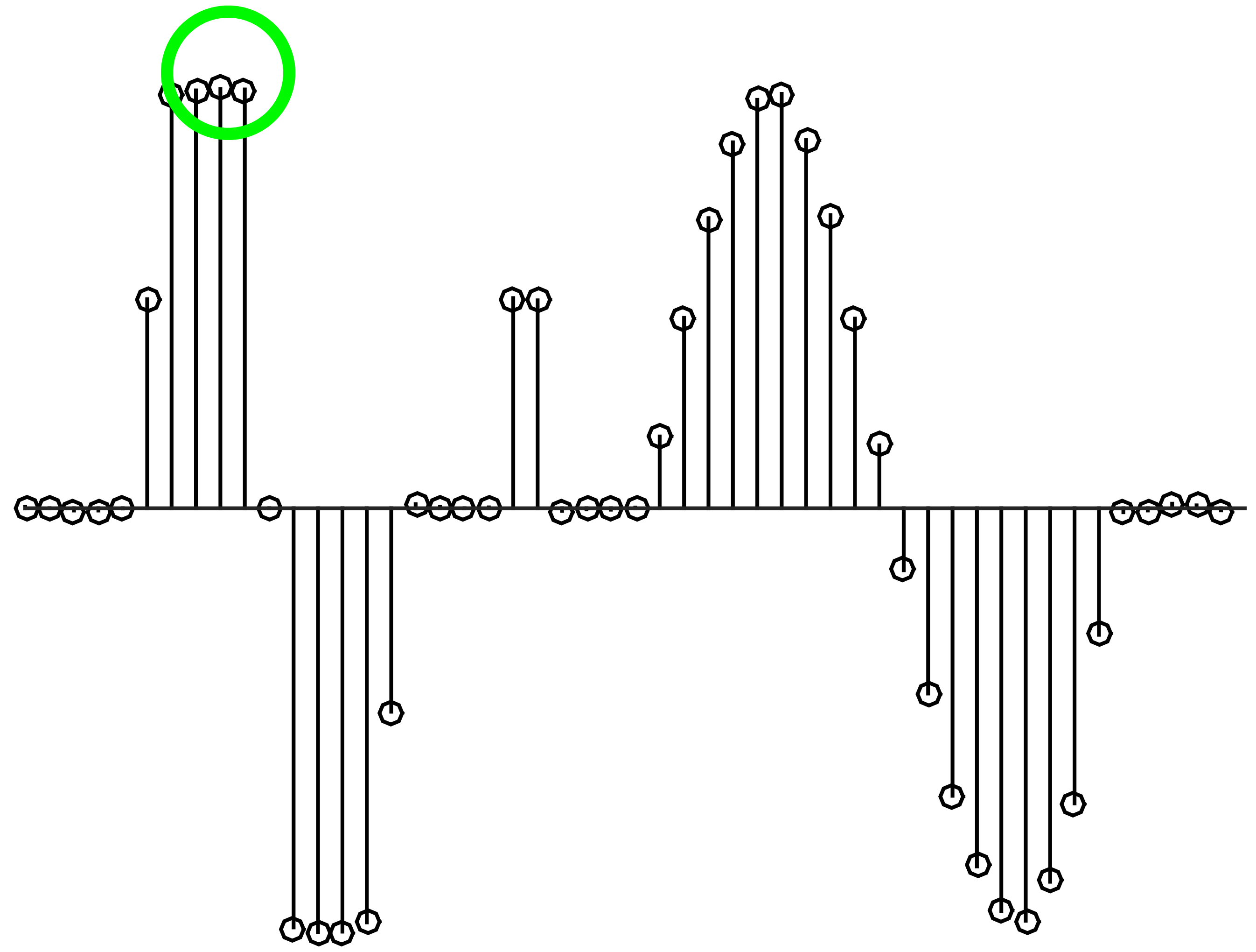


Example: Two-Point Moving Average

$x[t]$
$x[t-\Delta t]$

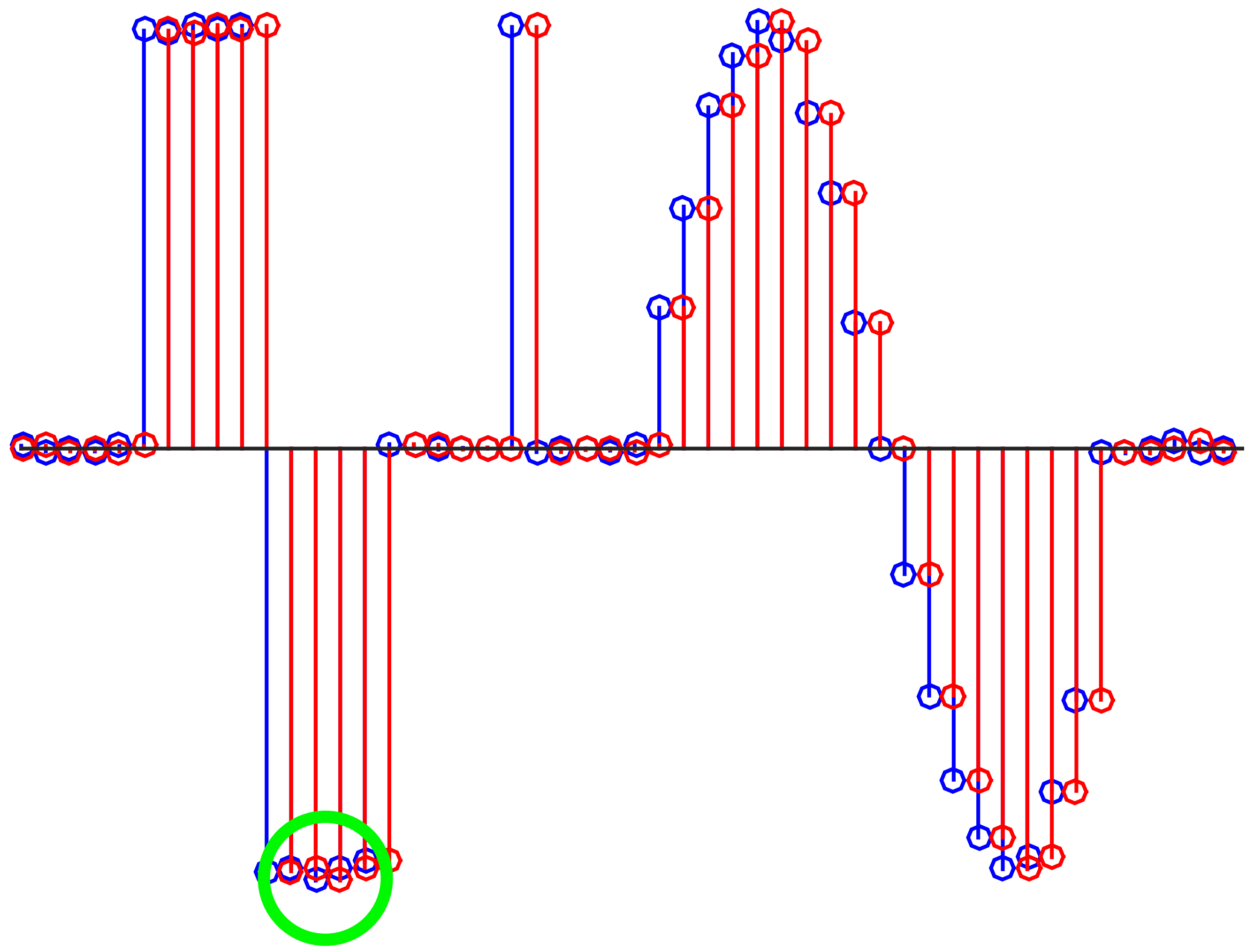


$$y[t]=\frac{1}{2}x[t]+\frac{1}{2}x[t-\Delta t]$$

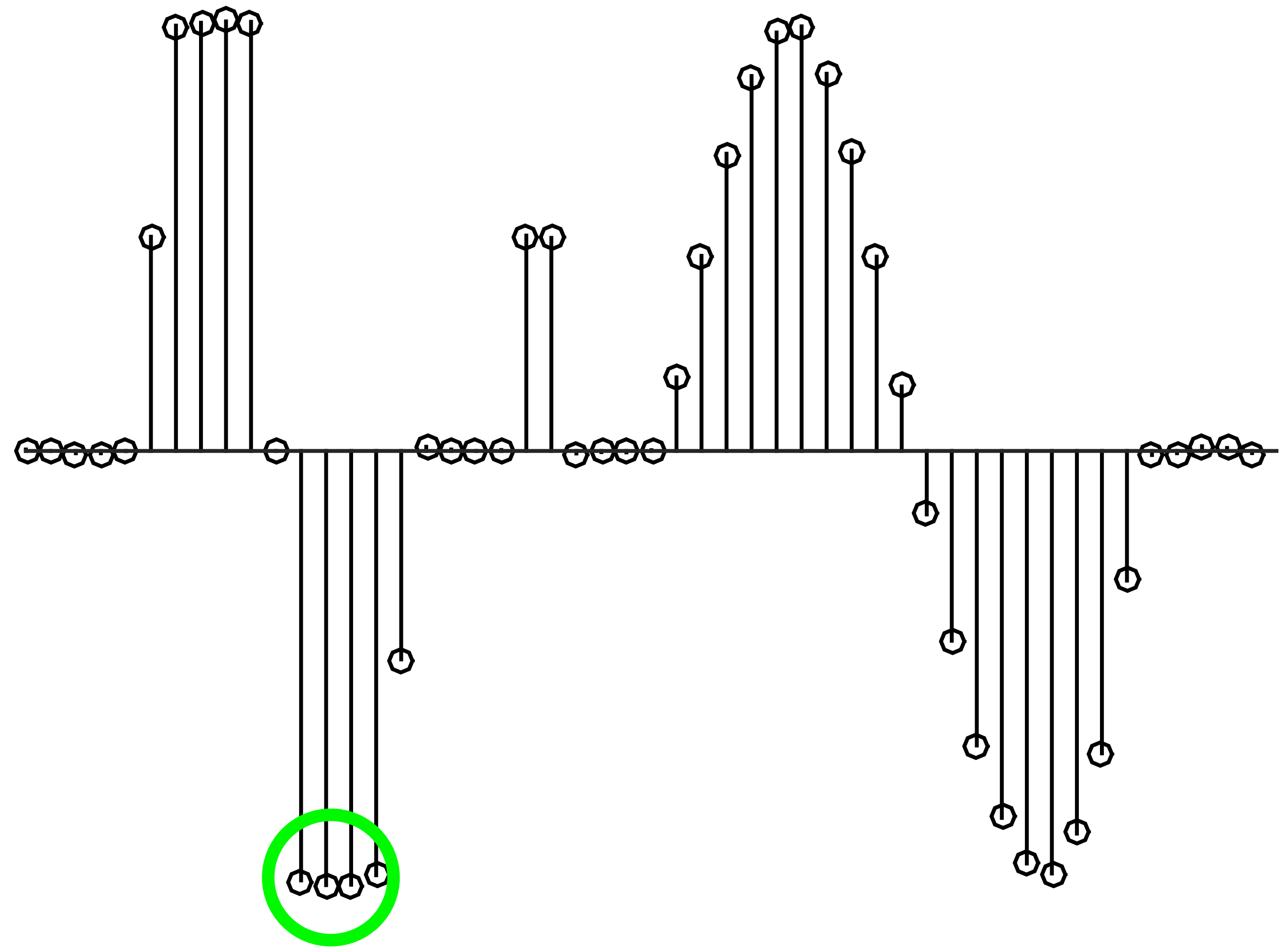


Example: Two-Point Moving Average

$x[t]$
$x[t-\Delta t]$

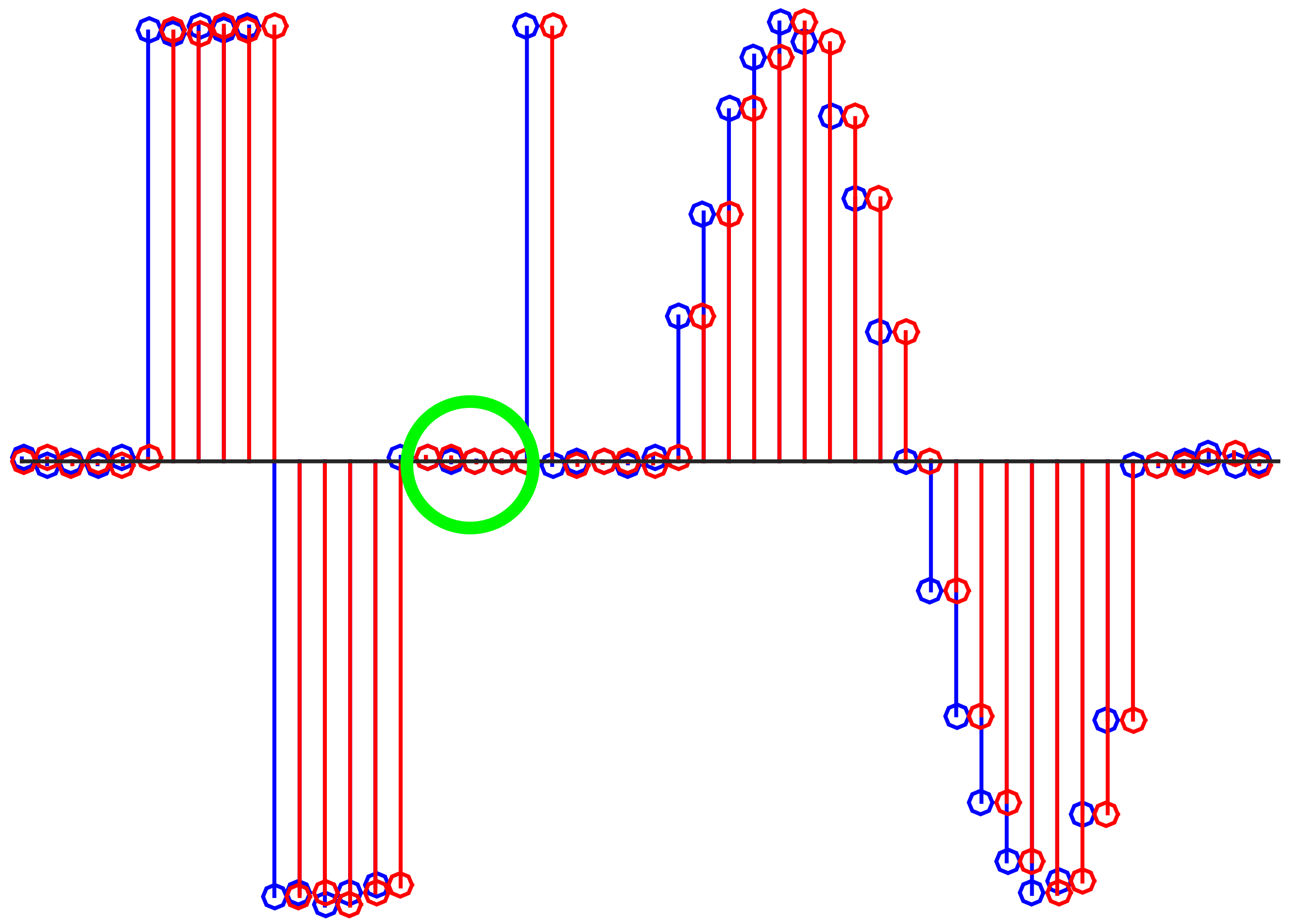


$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$

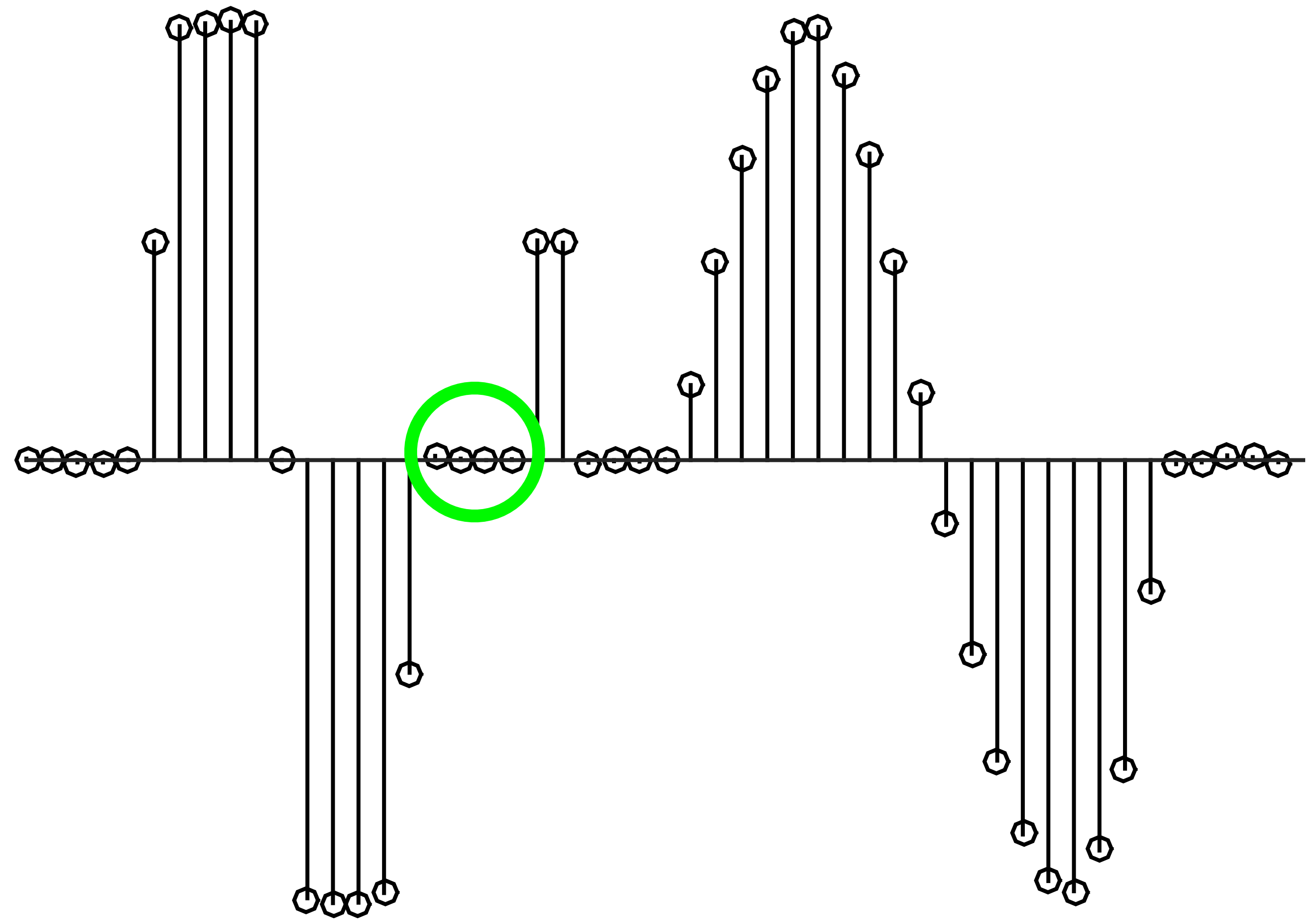


Example: Two-Point Moving Average

$x[t]$
$x[t-\Delta t]$

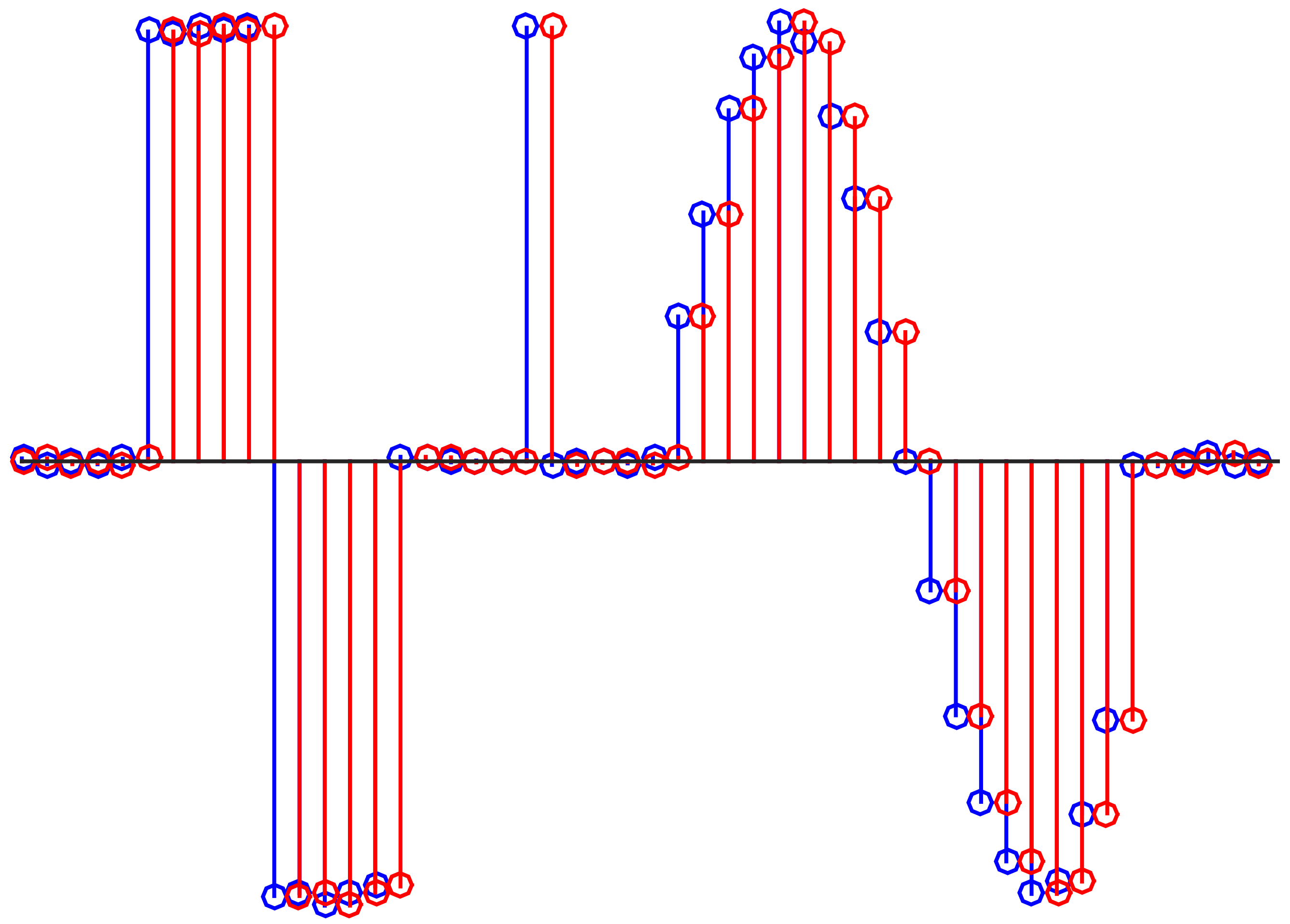


$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$

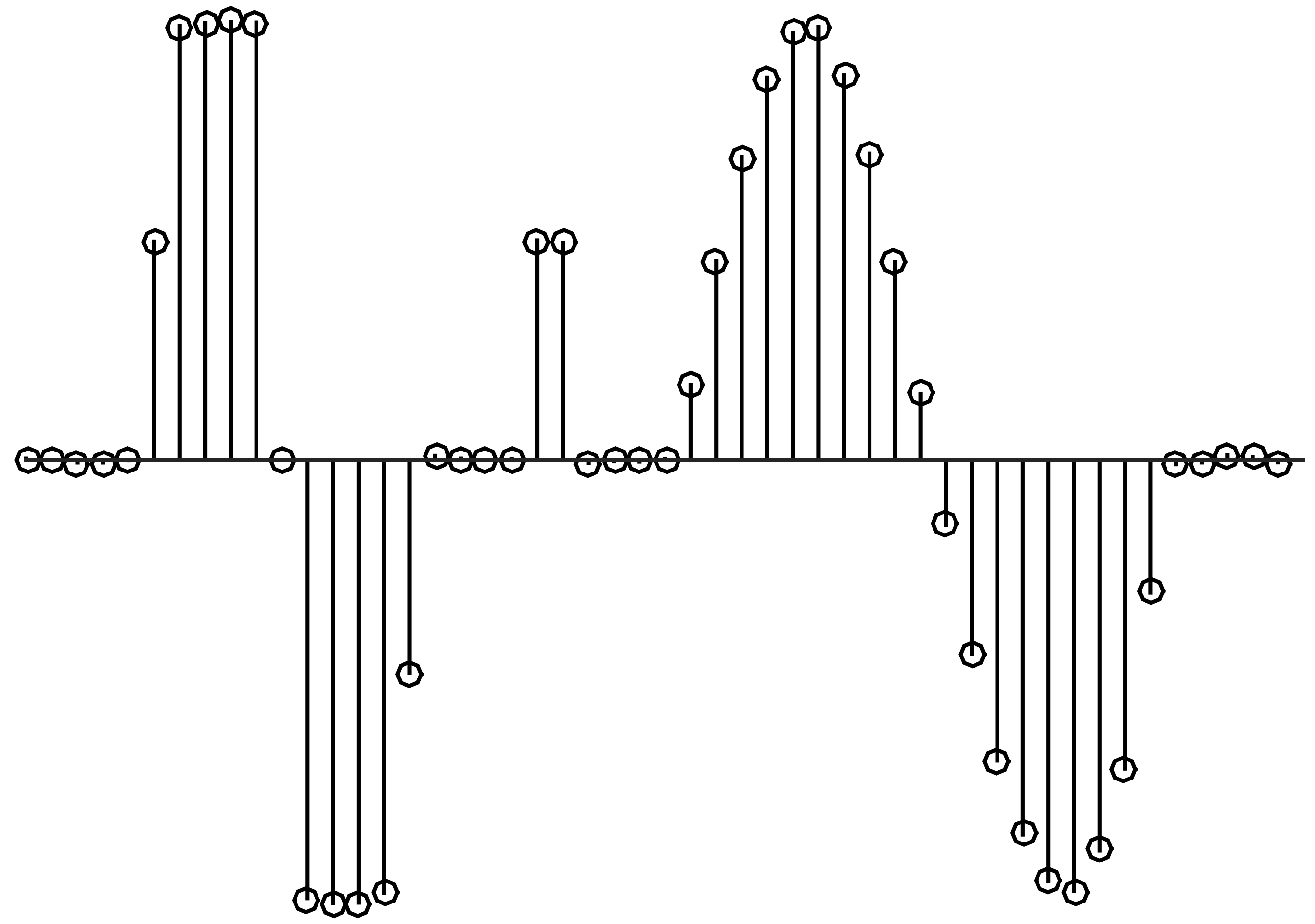


Example: Two-Point Moving Average

$$x[t]$$
$$x[t-\Delta t]$$

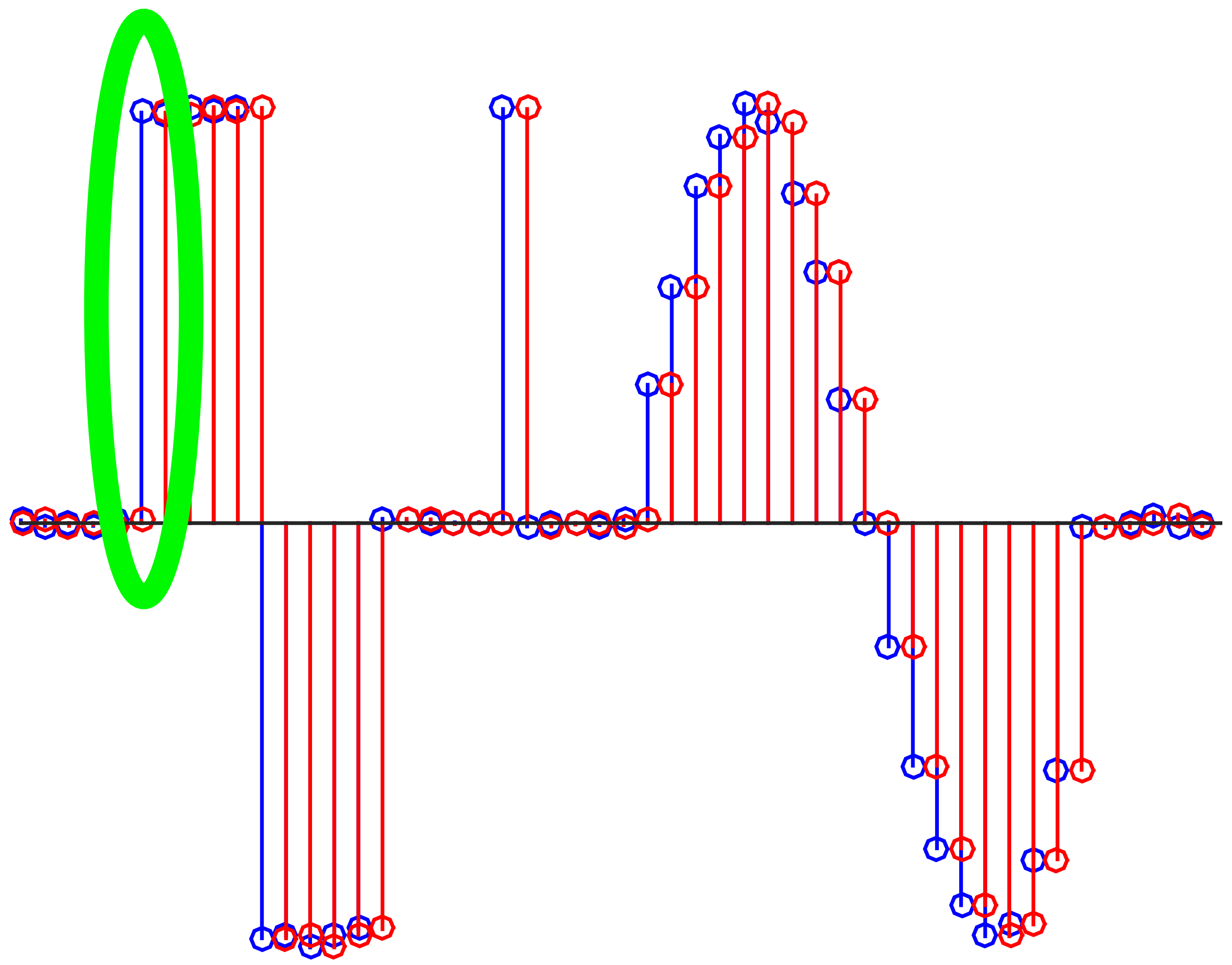


$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$

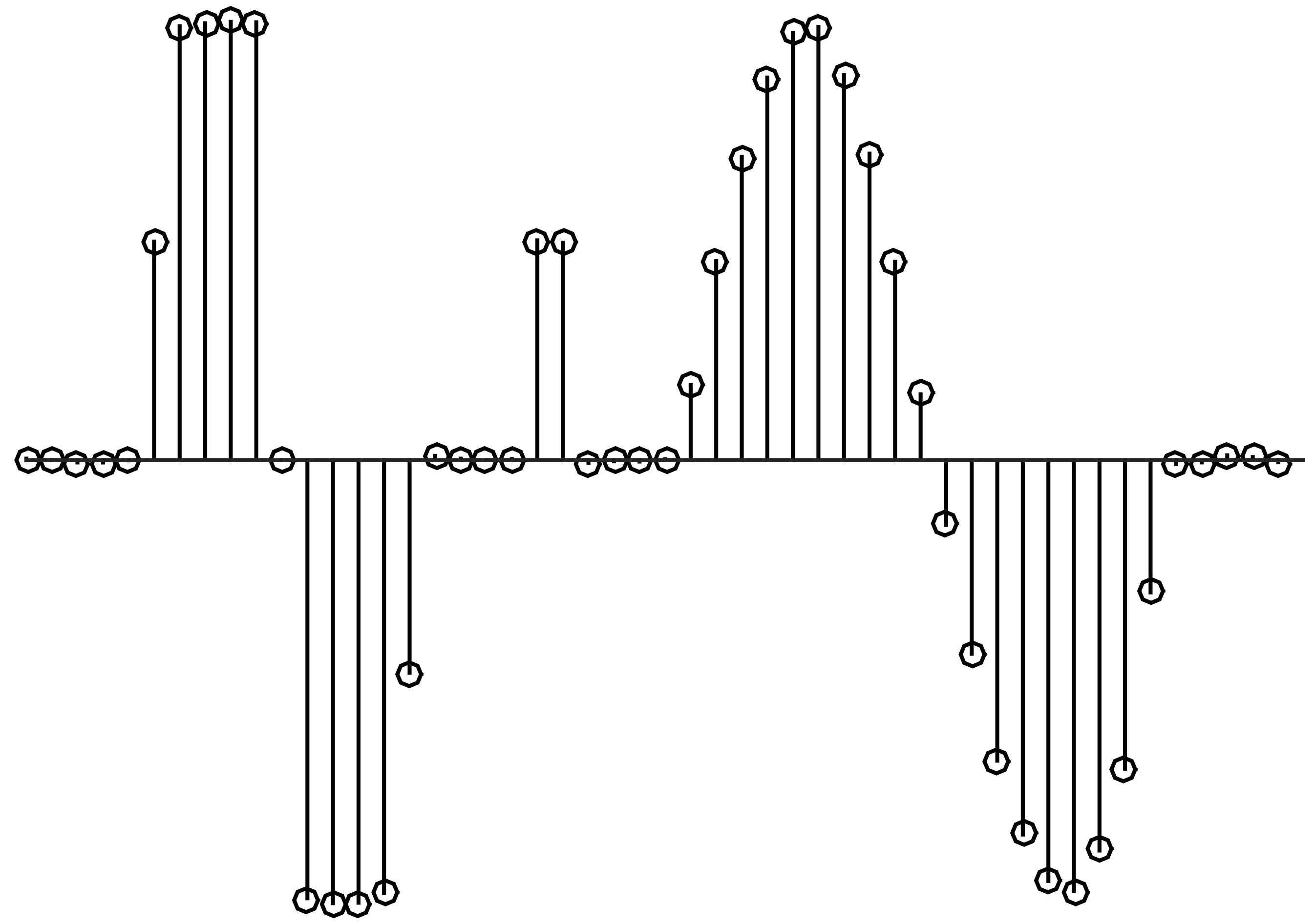


Example: Two-Point Moving Average

$$x[t]$$
$$x[t-\Delta t]$$

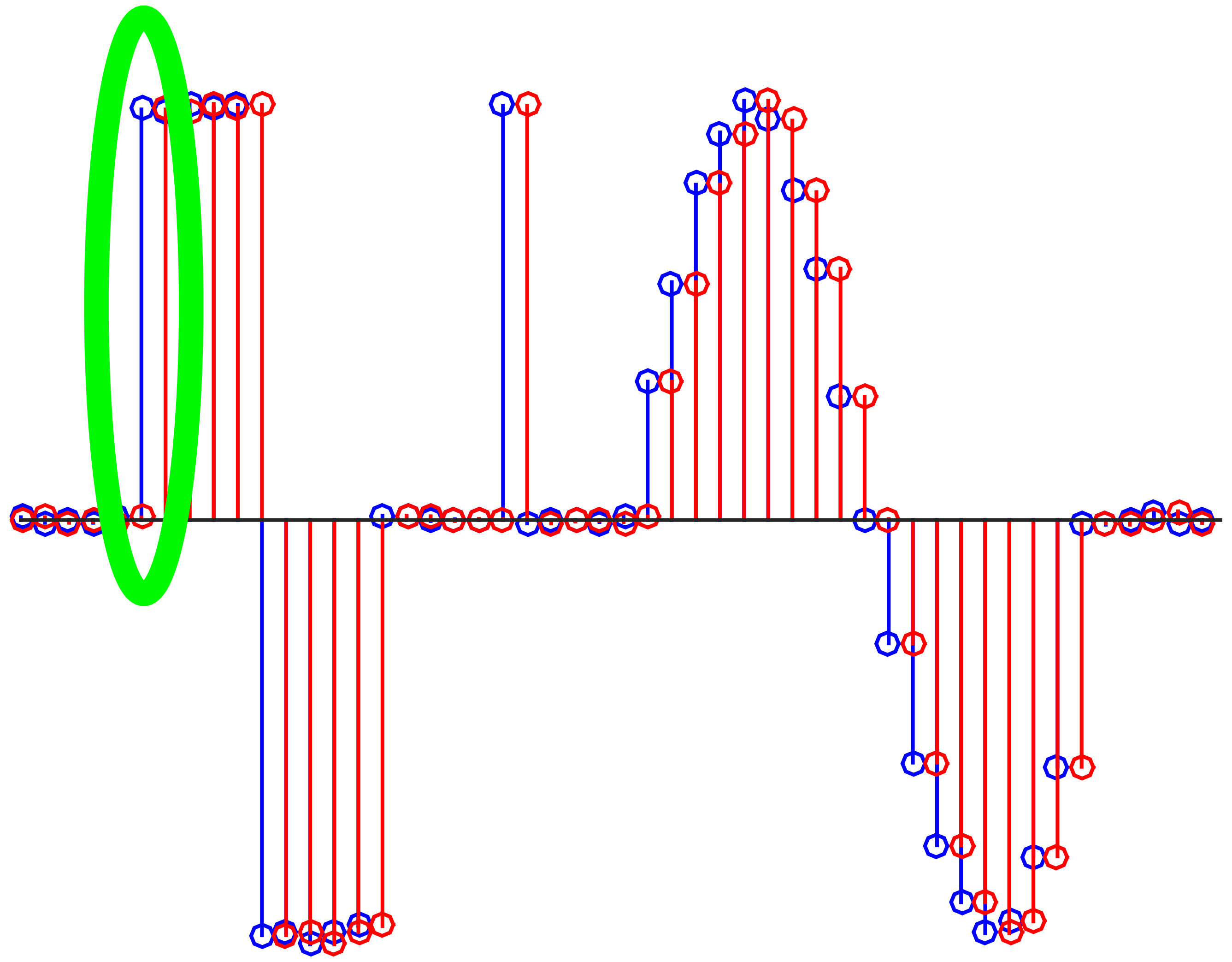


$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$

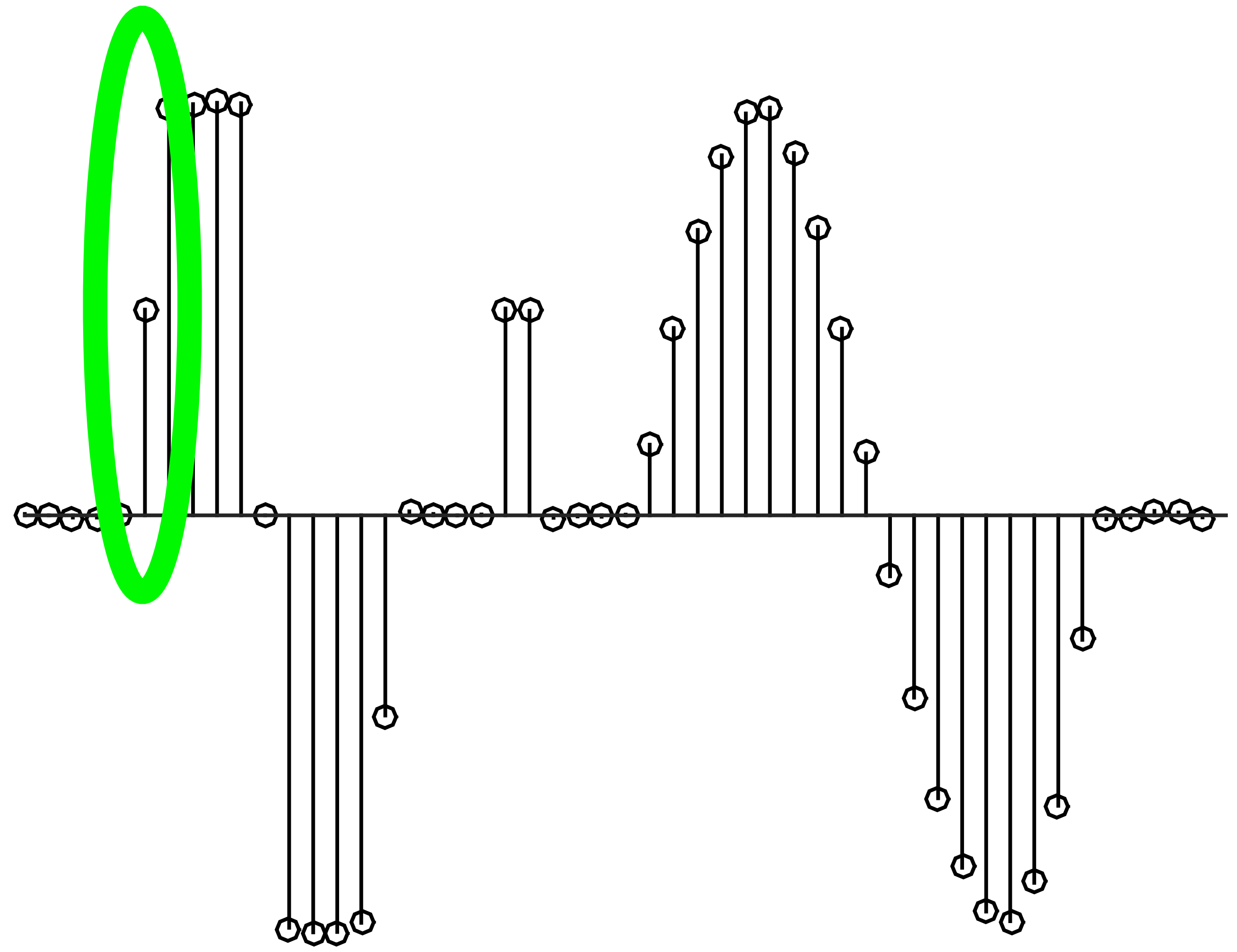


Example: Two-Point Moving Average

$$x[t]$$
$$x[t-\Delta t]$$

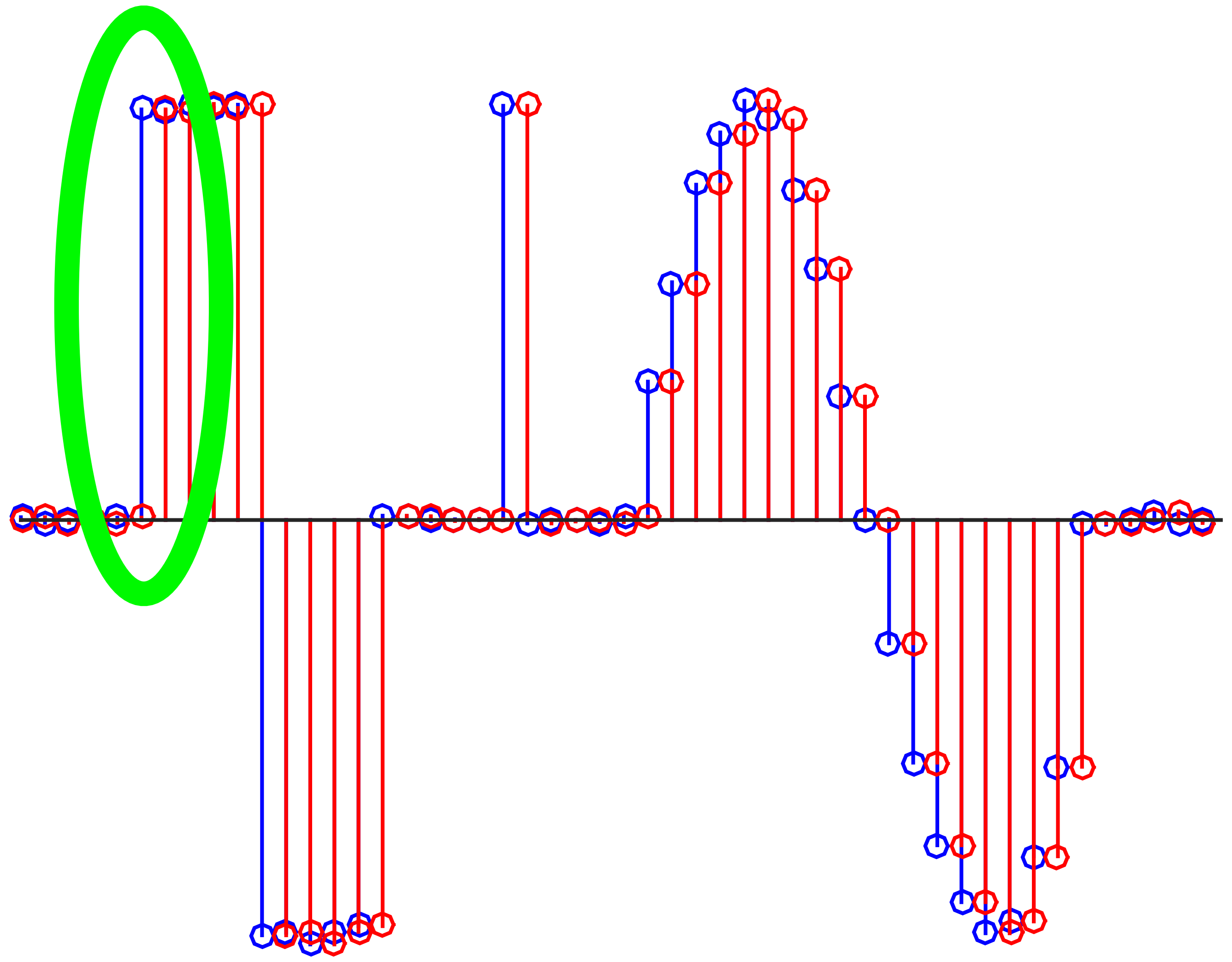


$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$

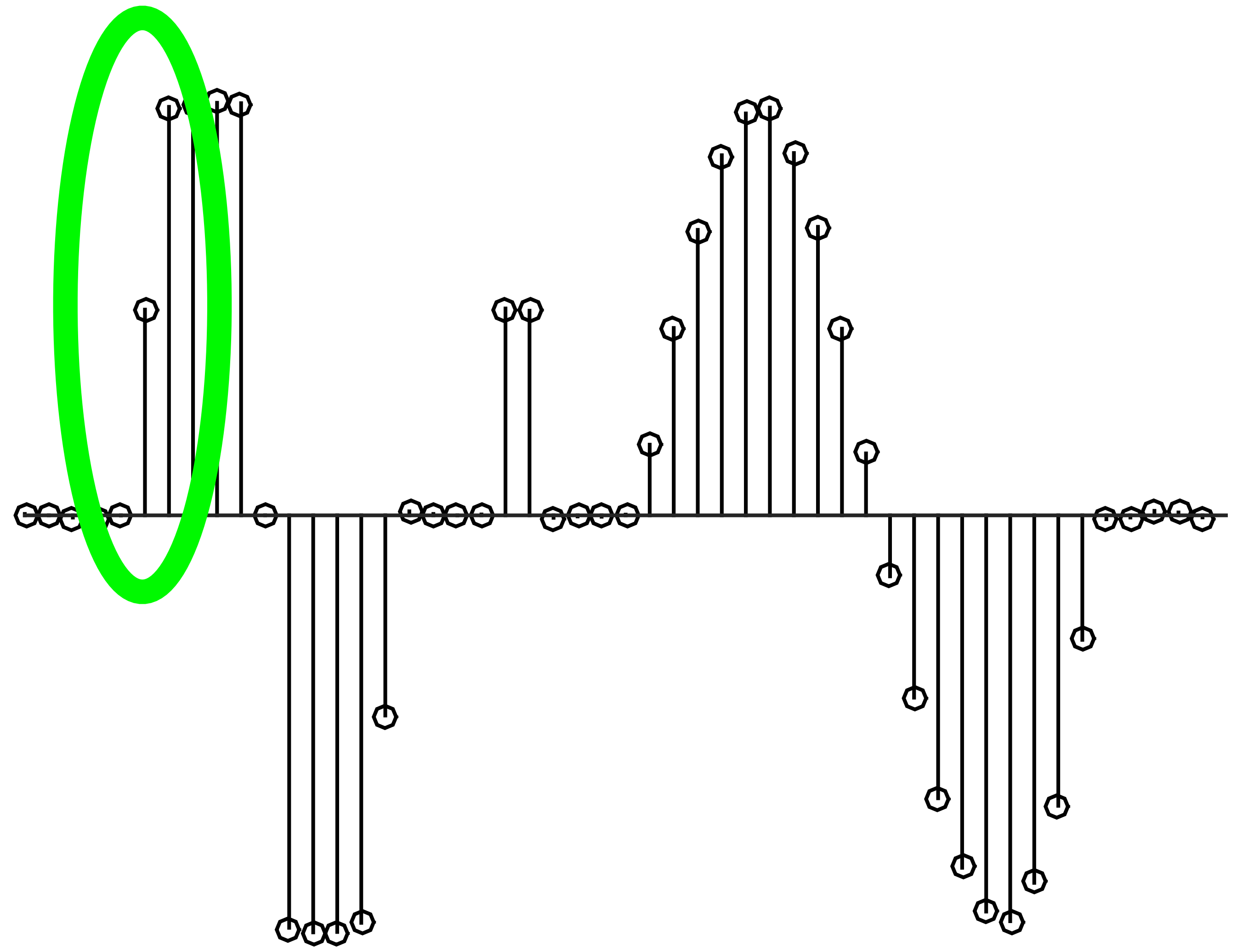


Example: Two-Point Moving Average

$x[t]$
$x[t-\Delta t]$

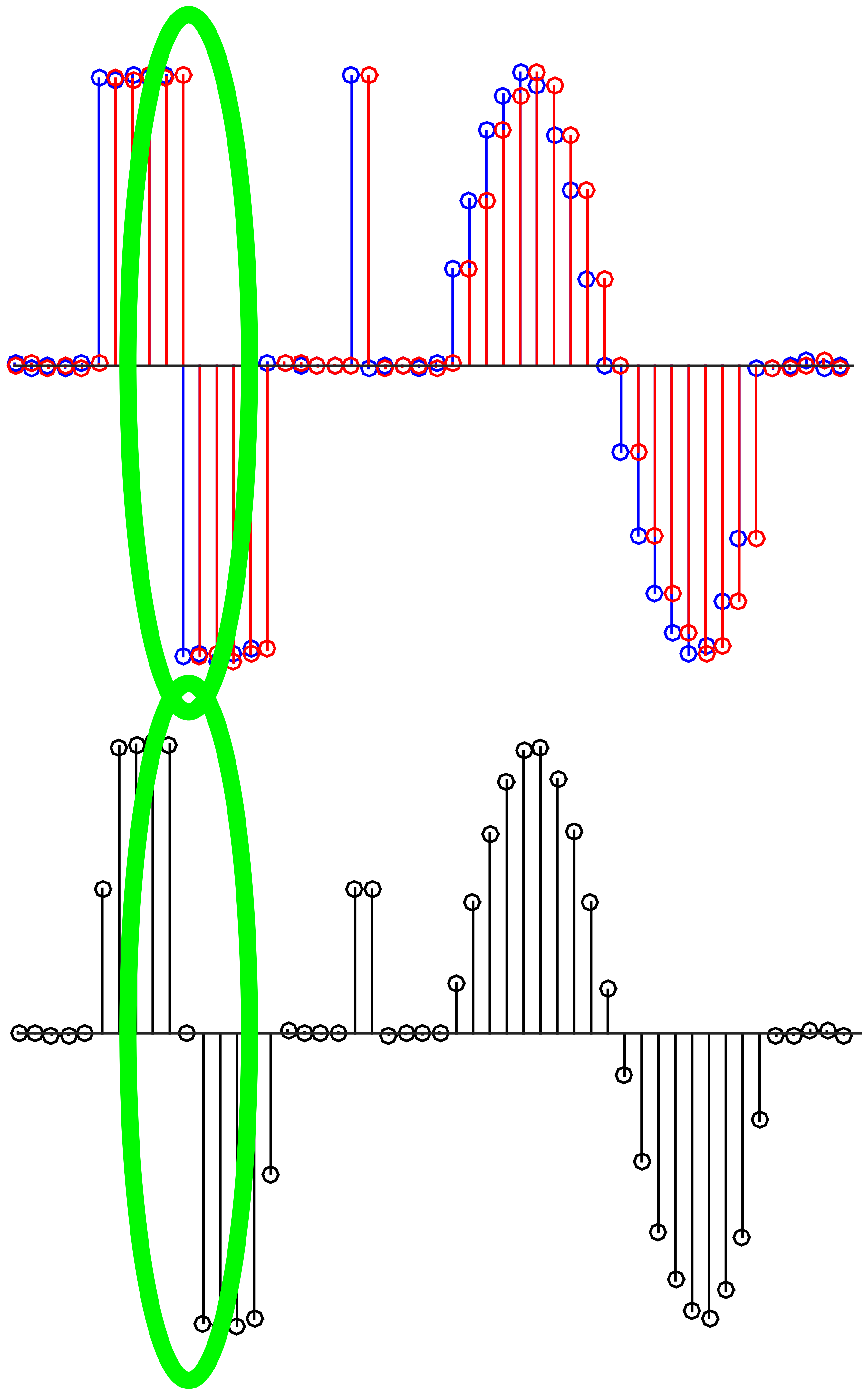


$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$



Example: Two-Point Moving Average

$x[t]$
$x[t-\Delta t]$

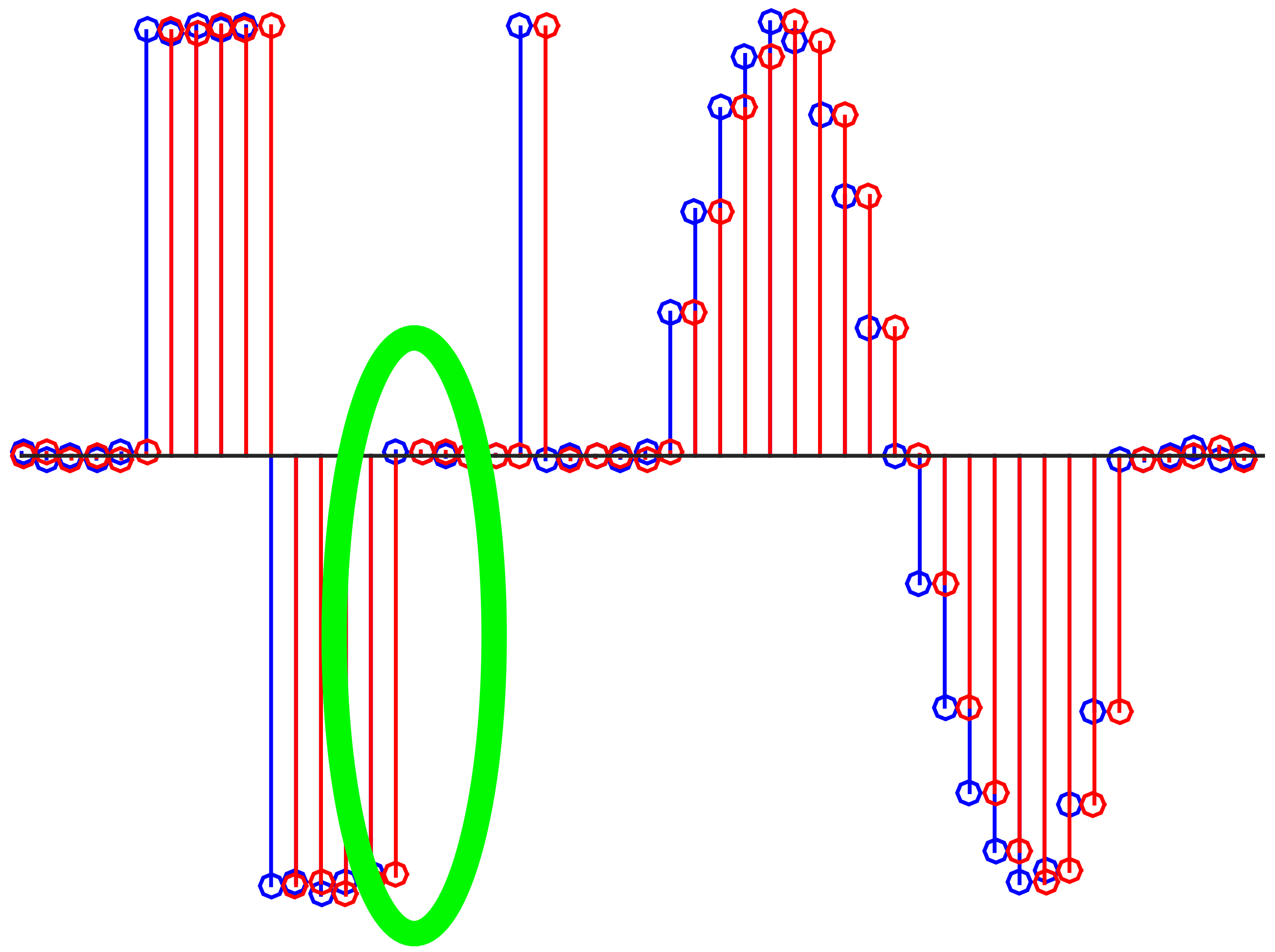


$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$

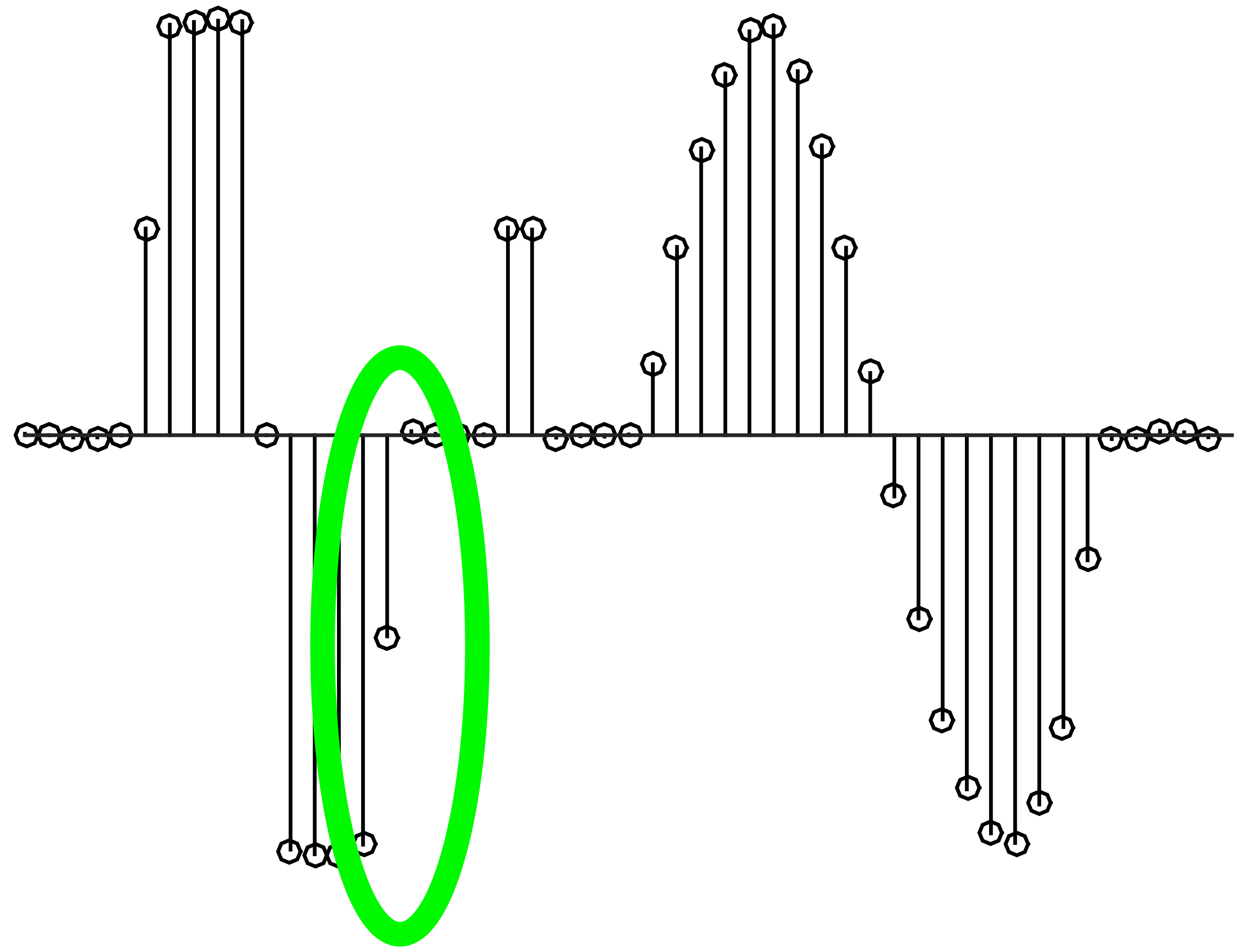
Example: Two-Point Moving Average

$x[t]$

$x[t-\Delta t]$

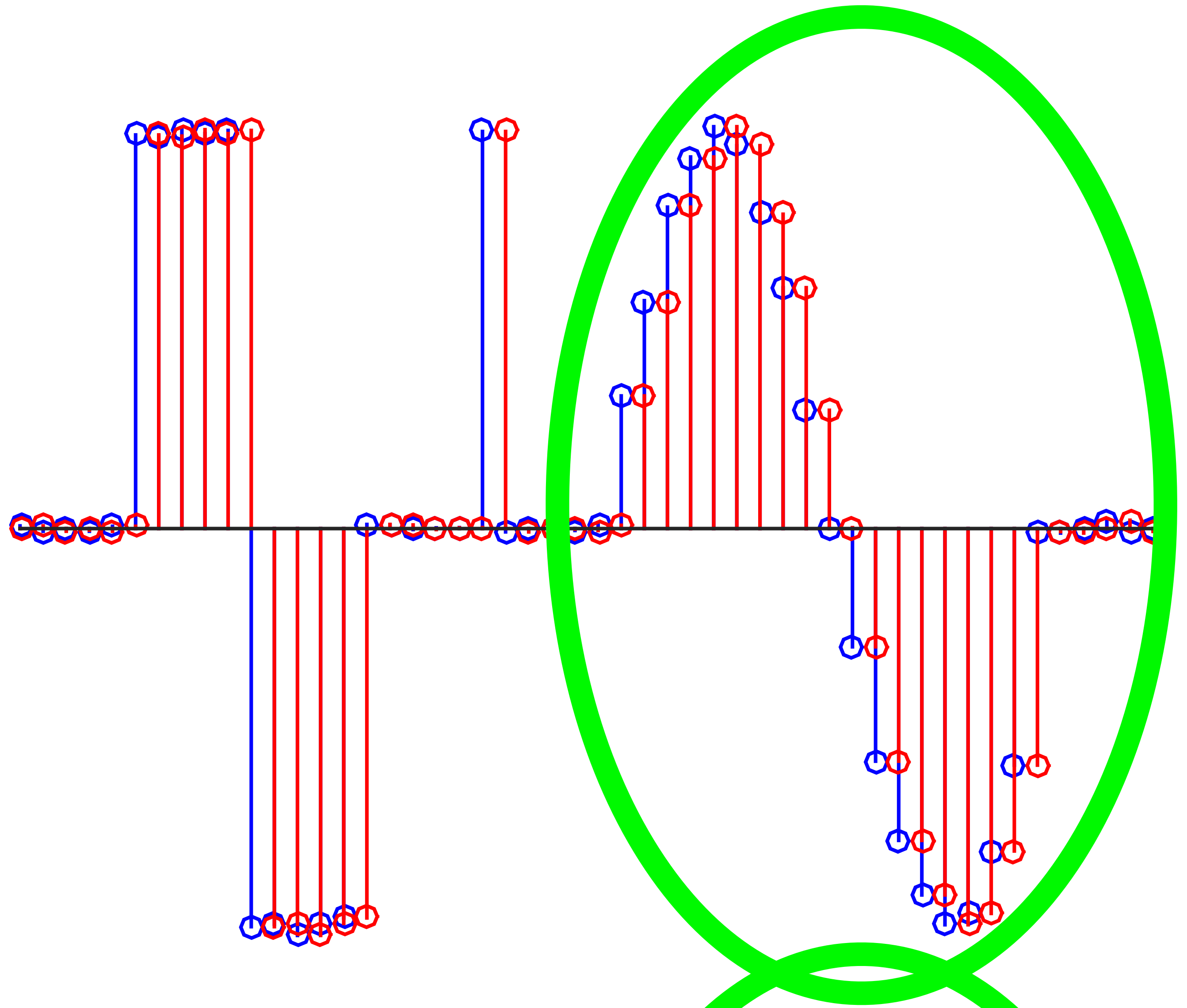


$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$

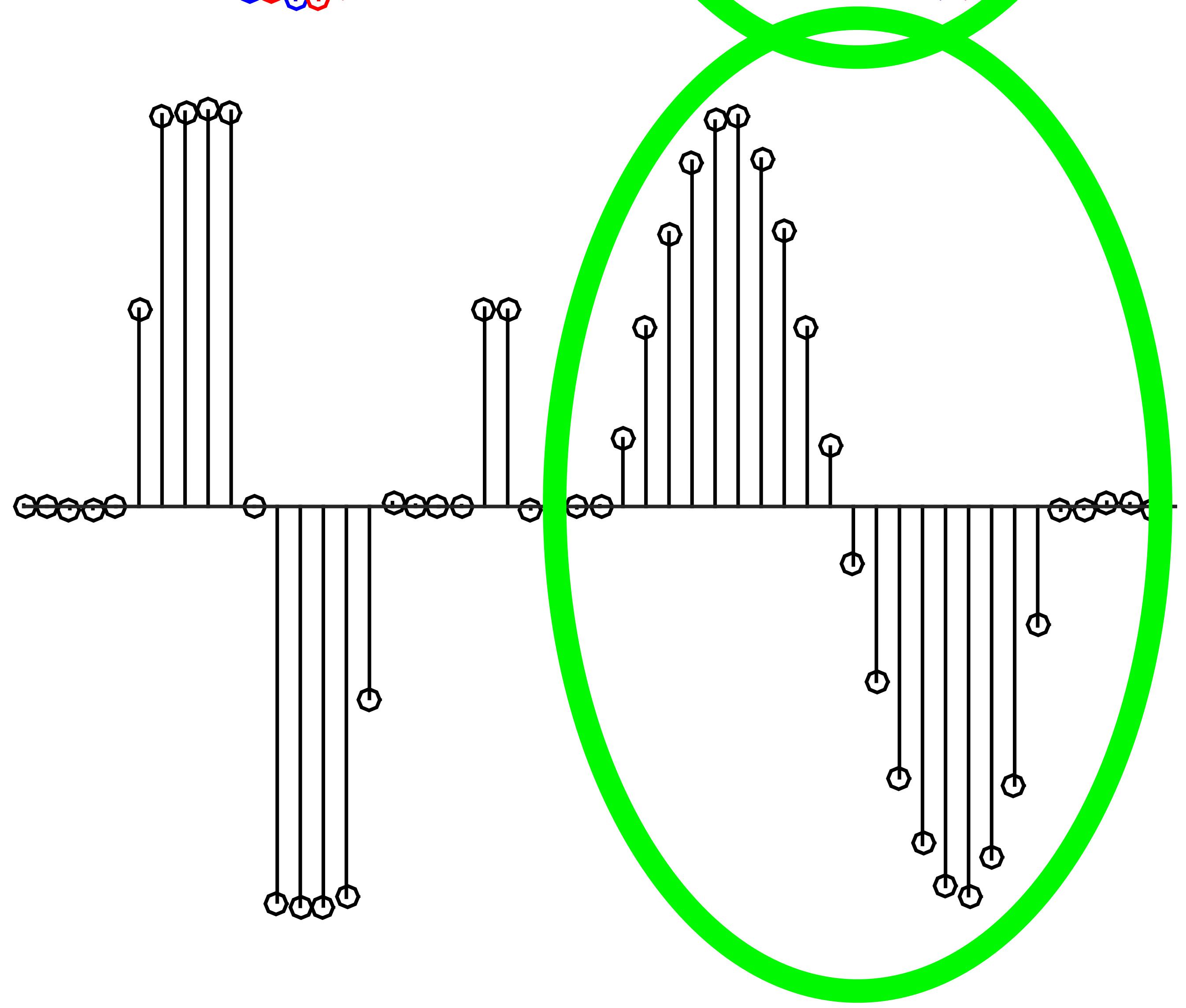


Example: Two-Point Moving Average

$x[t]$
$x[t-\Delta t]$

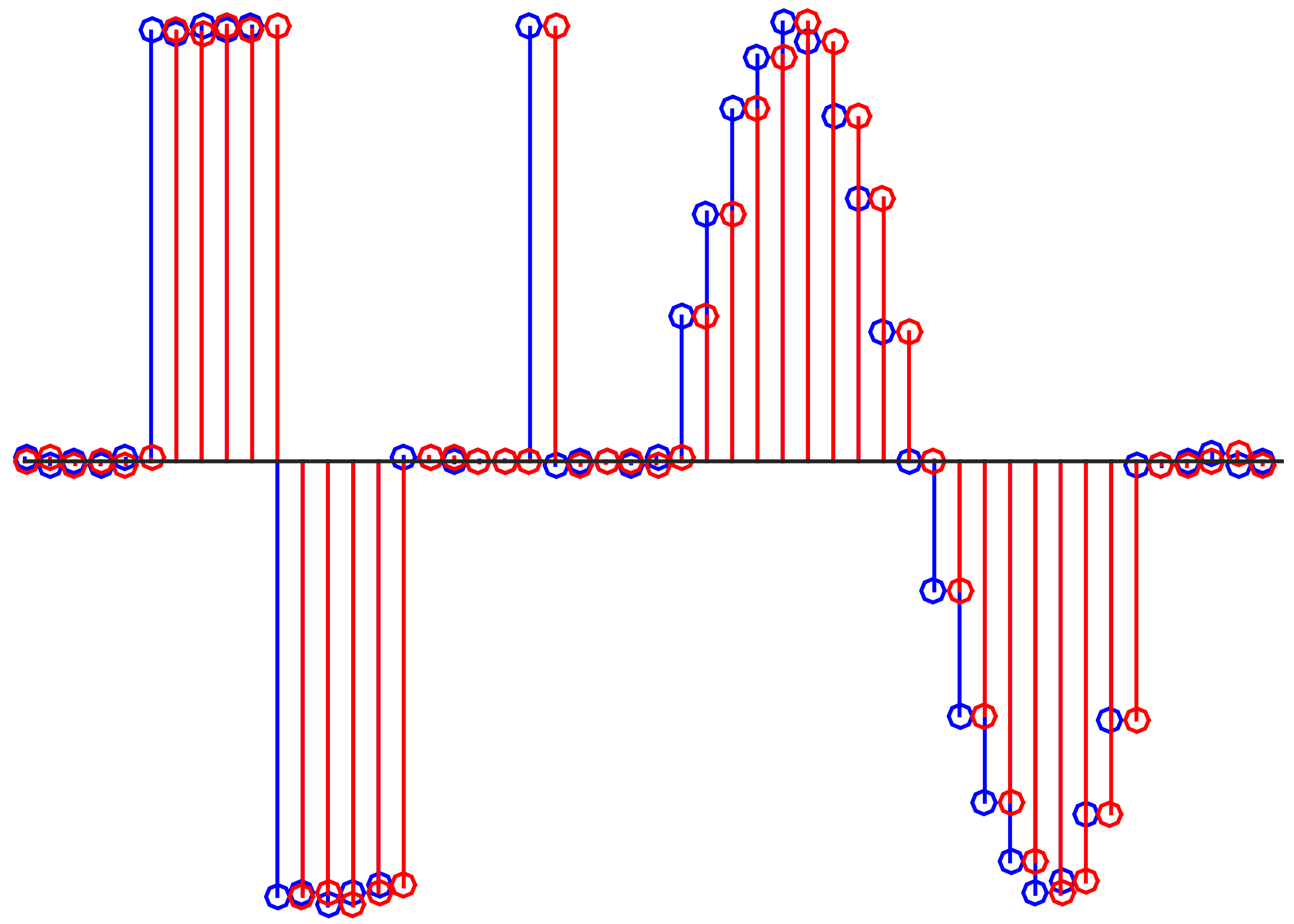


$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$

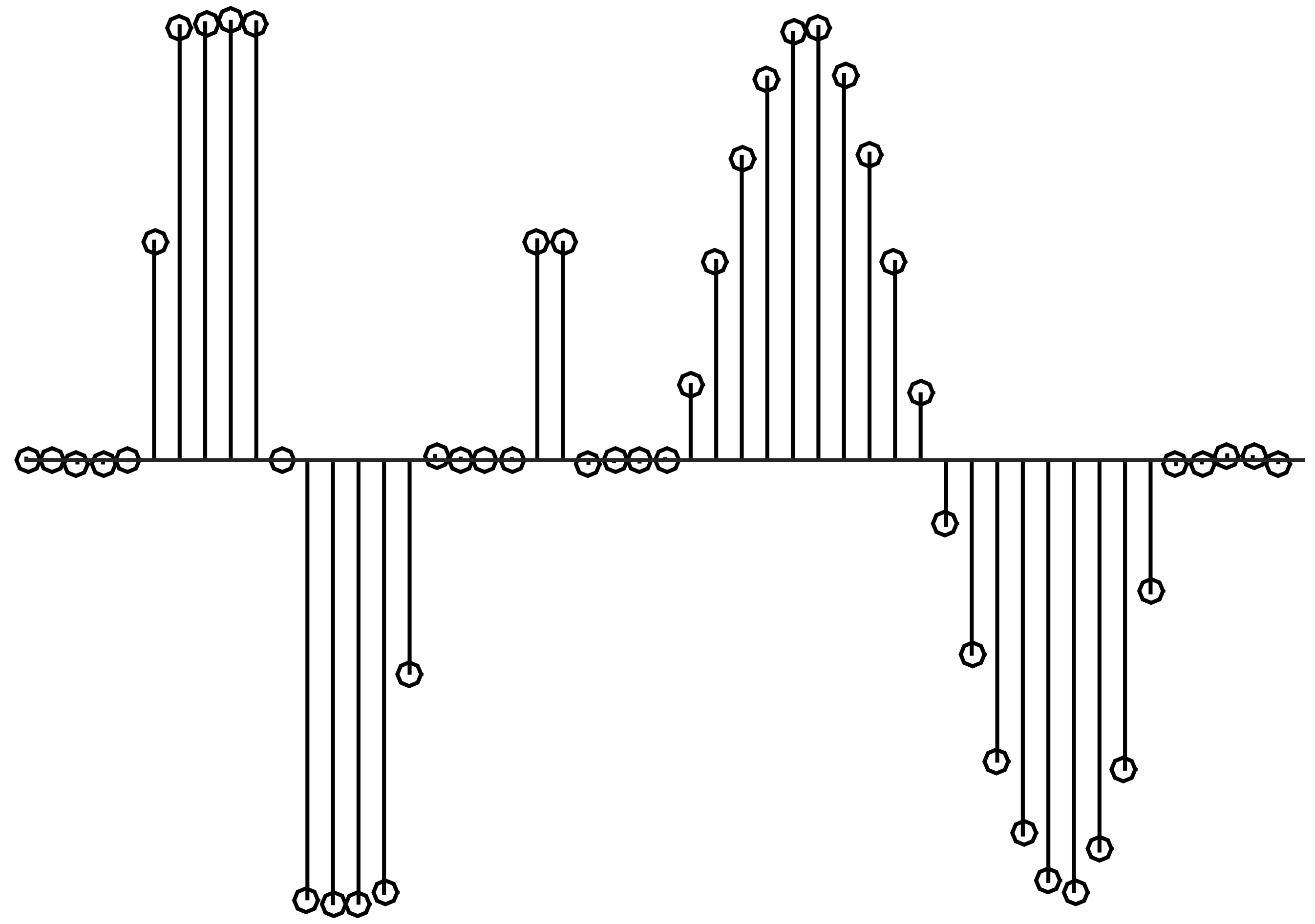


Example: Two-Point Moving Average

$$x[t]$$
$$x[t-\Delta t]$$

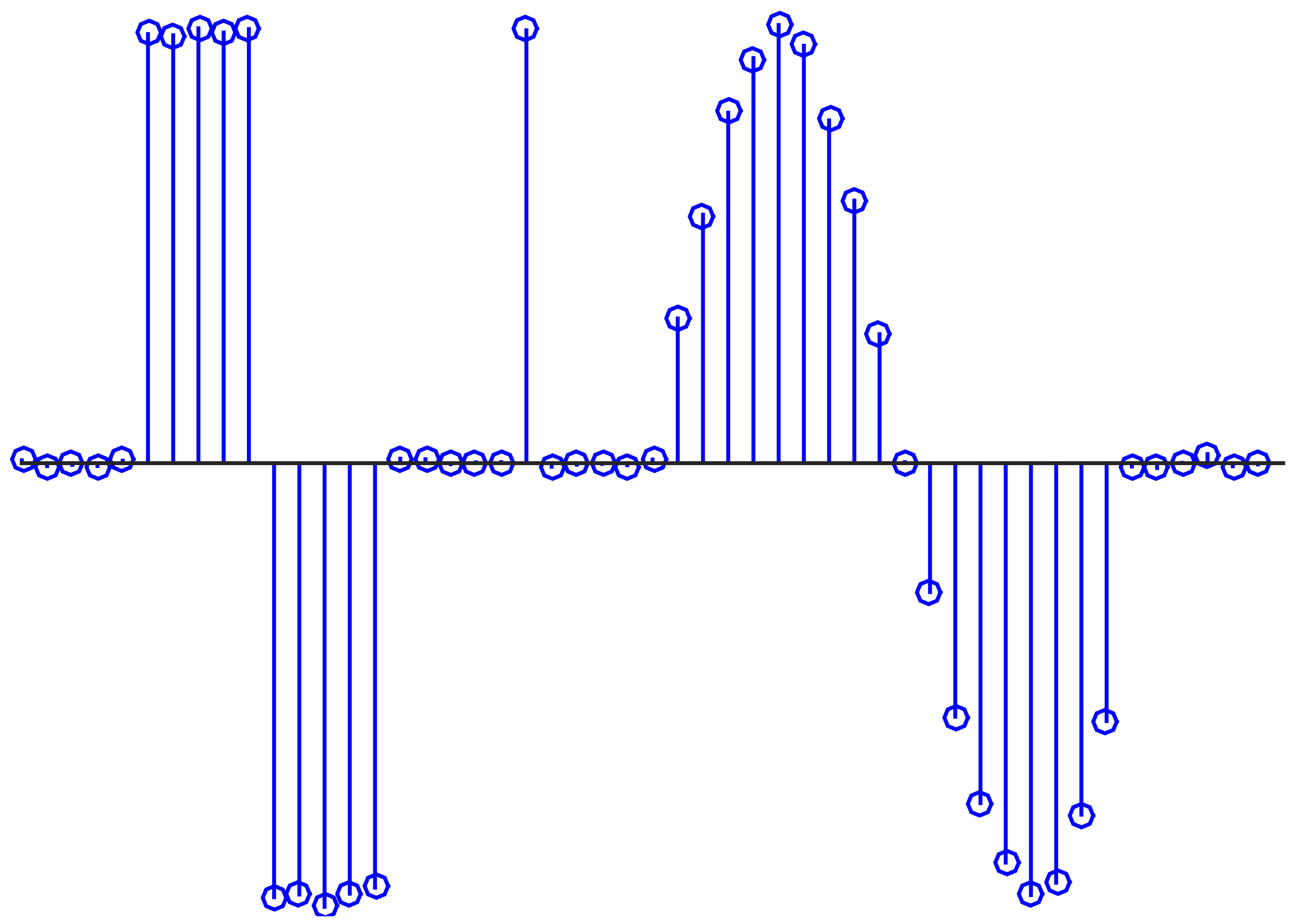


$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$

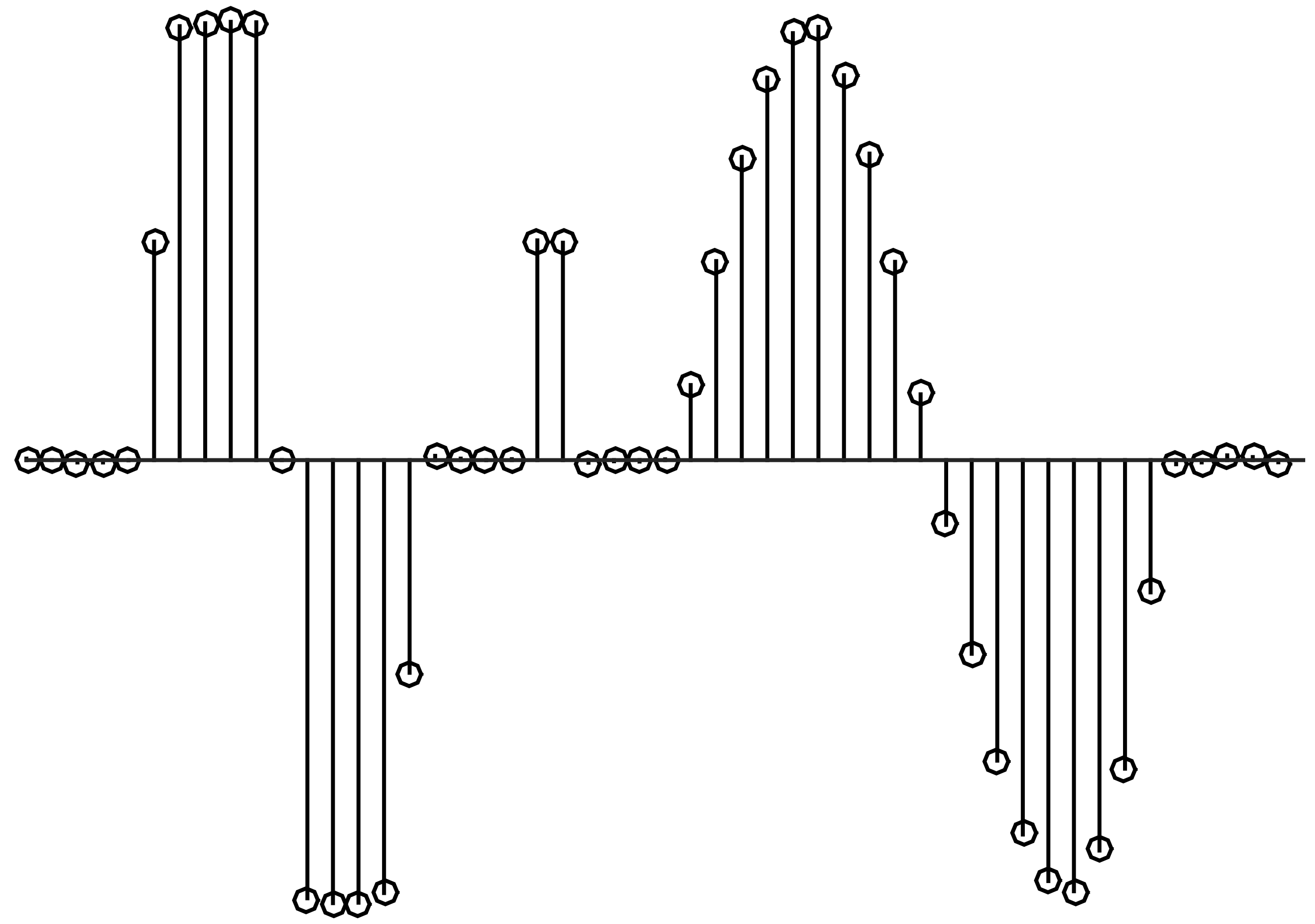


Example: Two-Point Moving Average

$x[t]$



$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t - \Delta t]$$

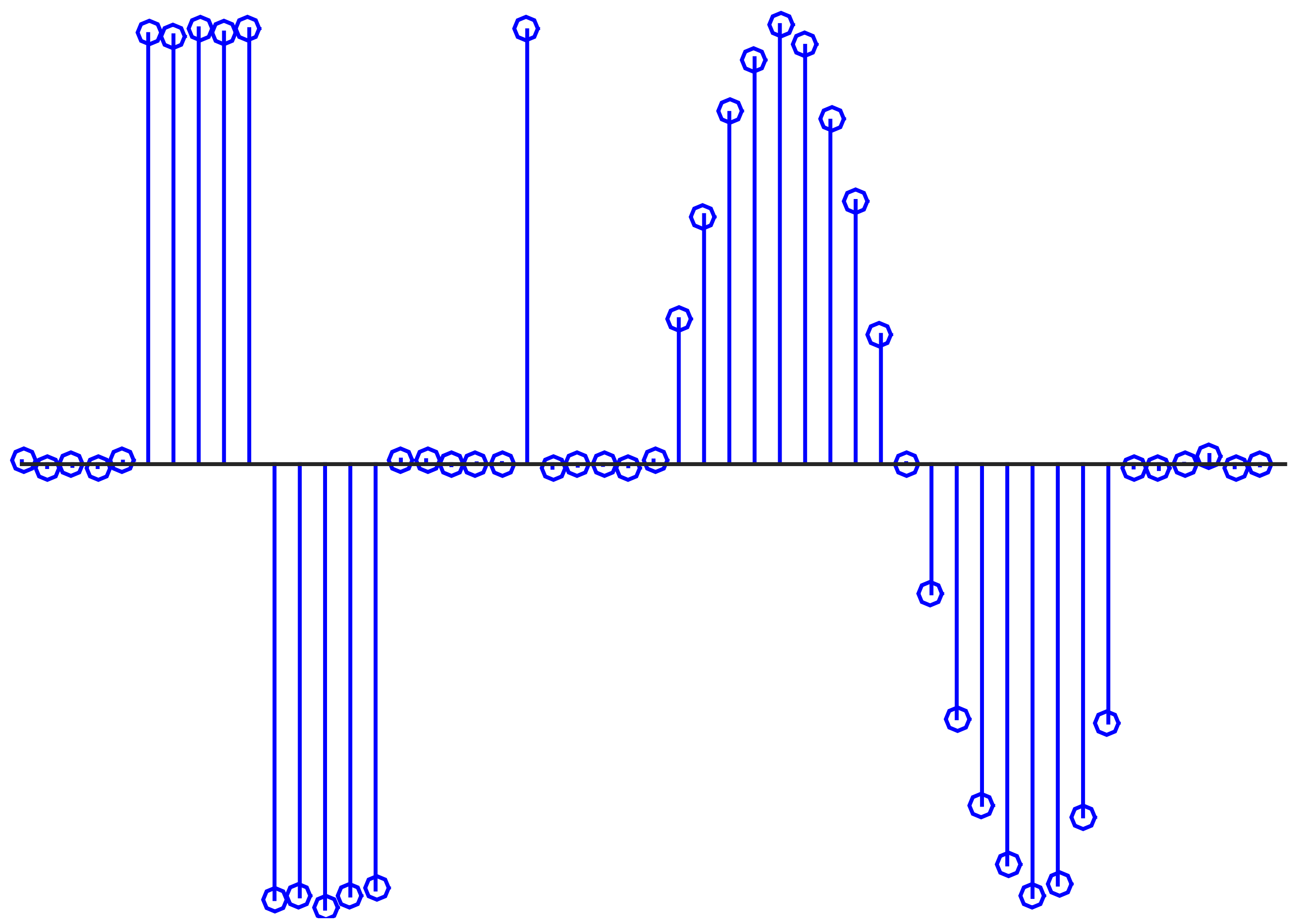


Results:

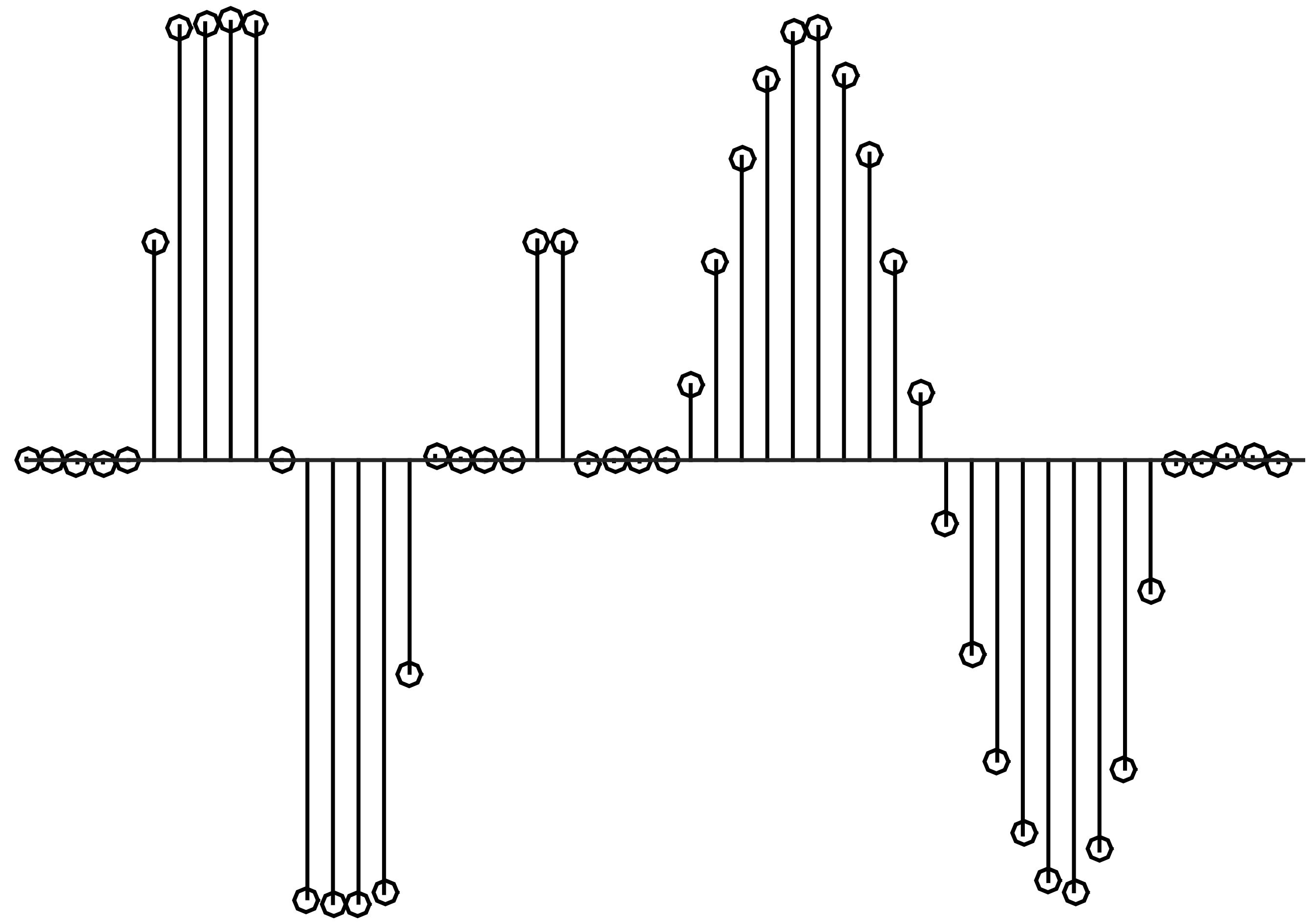
- Softens sudden changes
- Leaves slowly varying signals largely unchanged
- Slight delay in output relative to input
- Low Pass Filter?

Example: Two-Point Moving Average

$x[t]$



$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t - \Delta t]$$



Results:

- Softens sudden changes
- Leaves slowly varying signals largely unchanged
- Slight delay in output relative to input
- Low Pass Filter?

Example: Two-Point Moving Difference

$$y[t] = \frac{x[t] - x[t - \Delta t]}{2}$$

What to Expect:

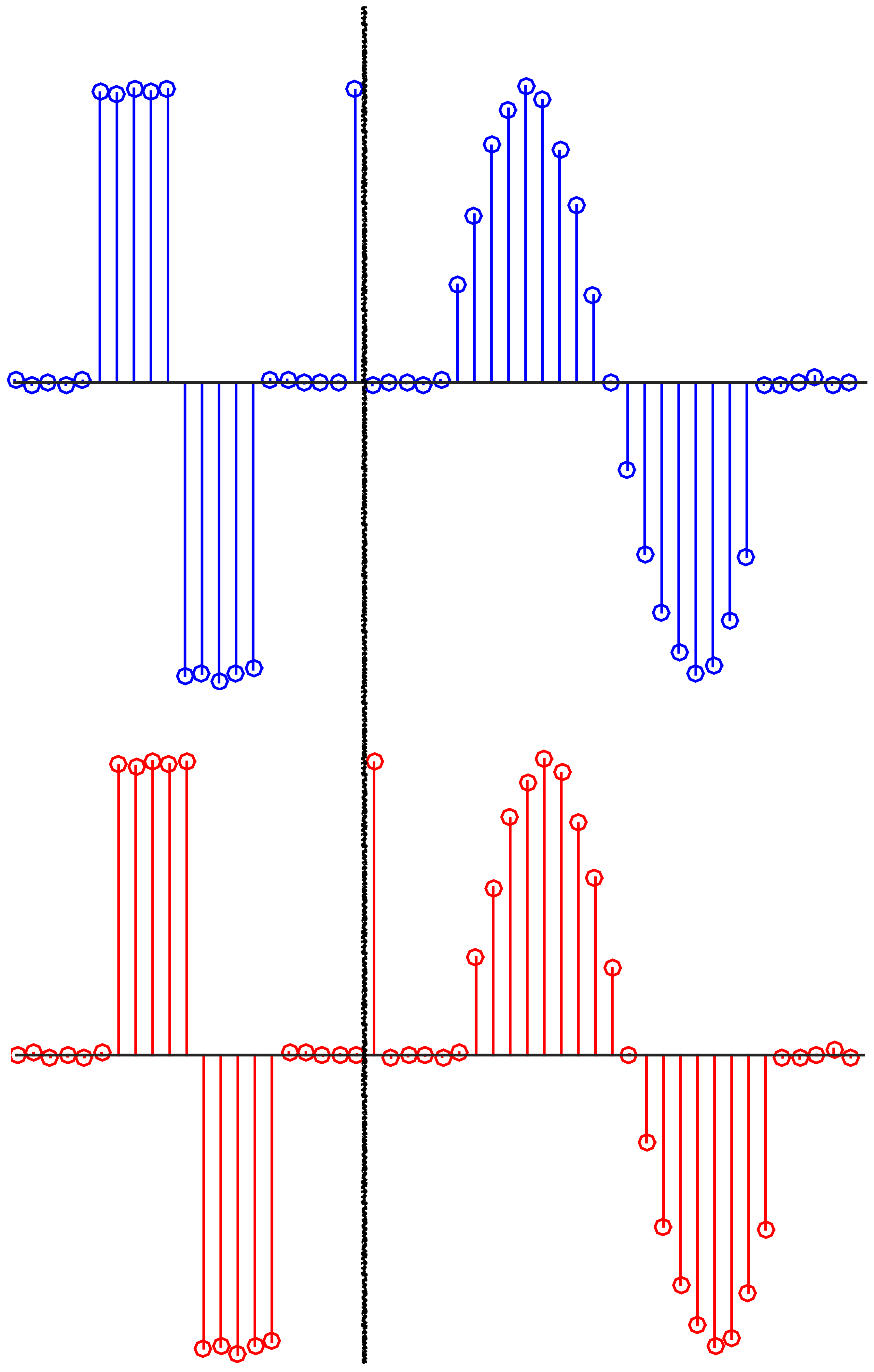
- Exaggerate differences
- Amplify quickly varying signals
- Attenuate slowly varying signals
- High Pass Filter?

Example: Two-Point Moving Difference

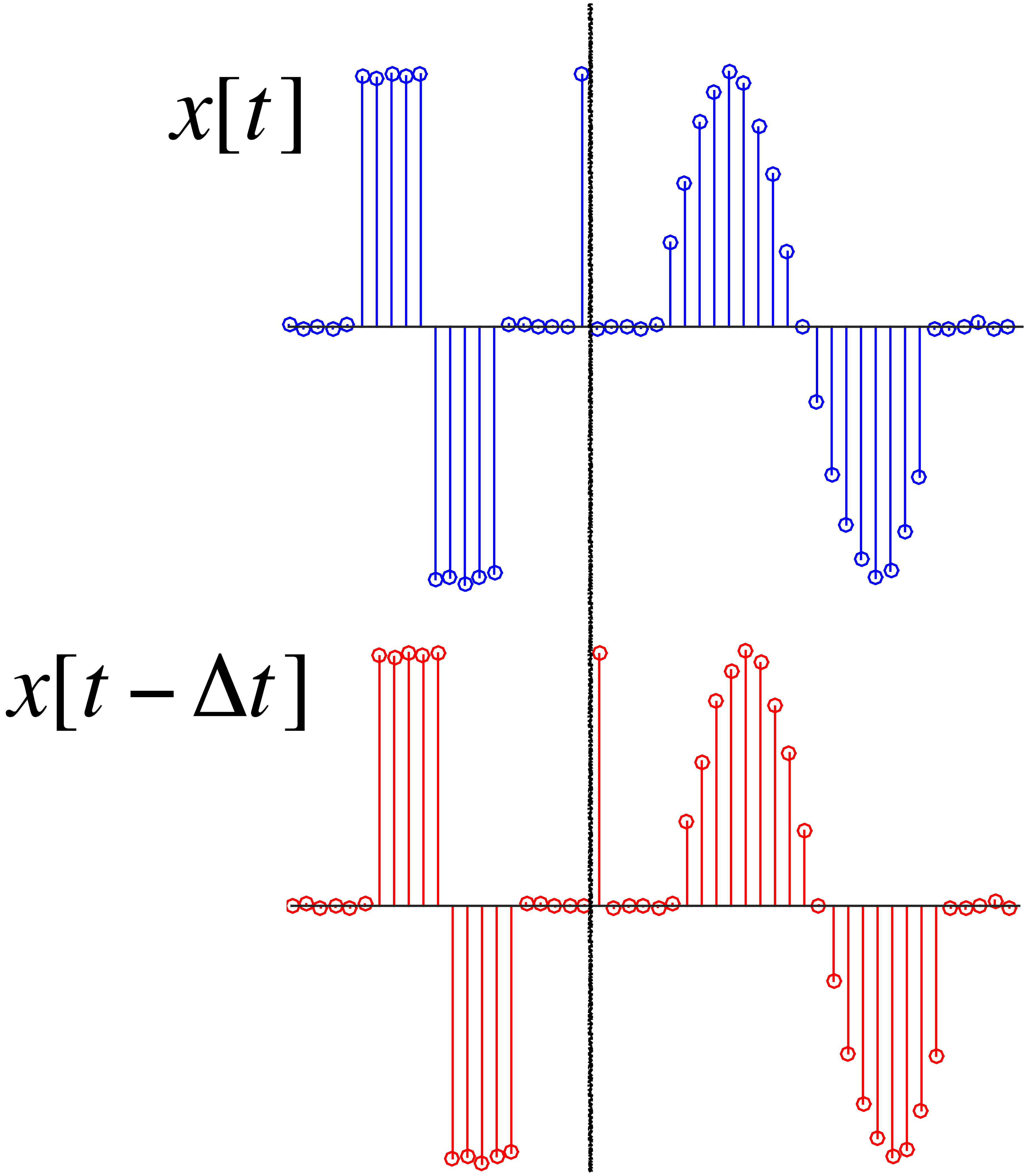
$$y[t] = \frac{1}{2}x[t] - \frac{1}{2}x[t - \Delta t]$$

$x[t]$

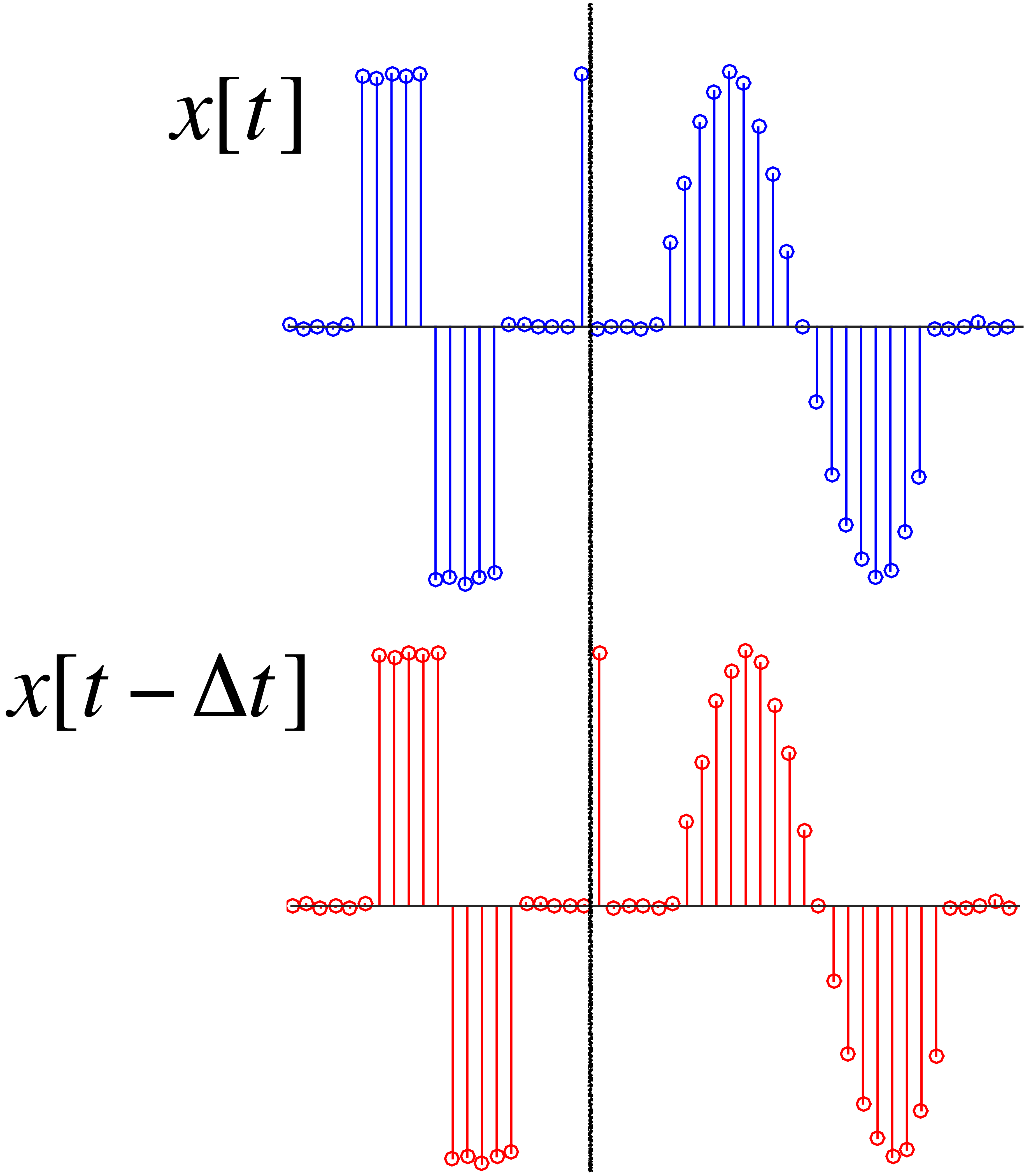
$x[t - \Delta t]$

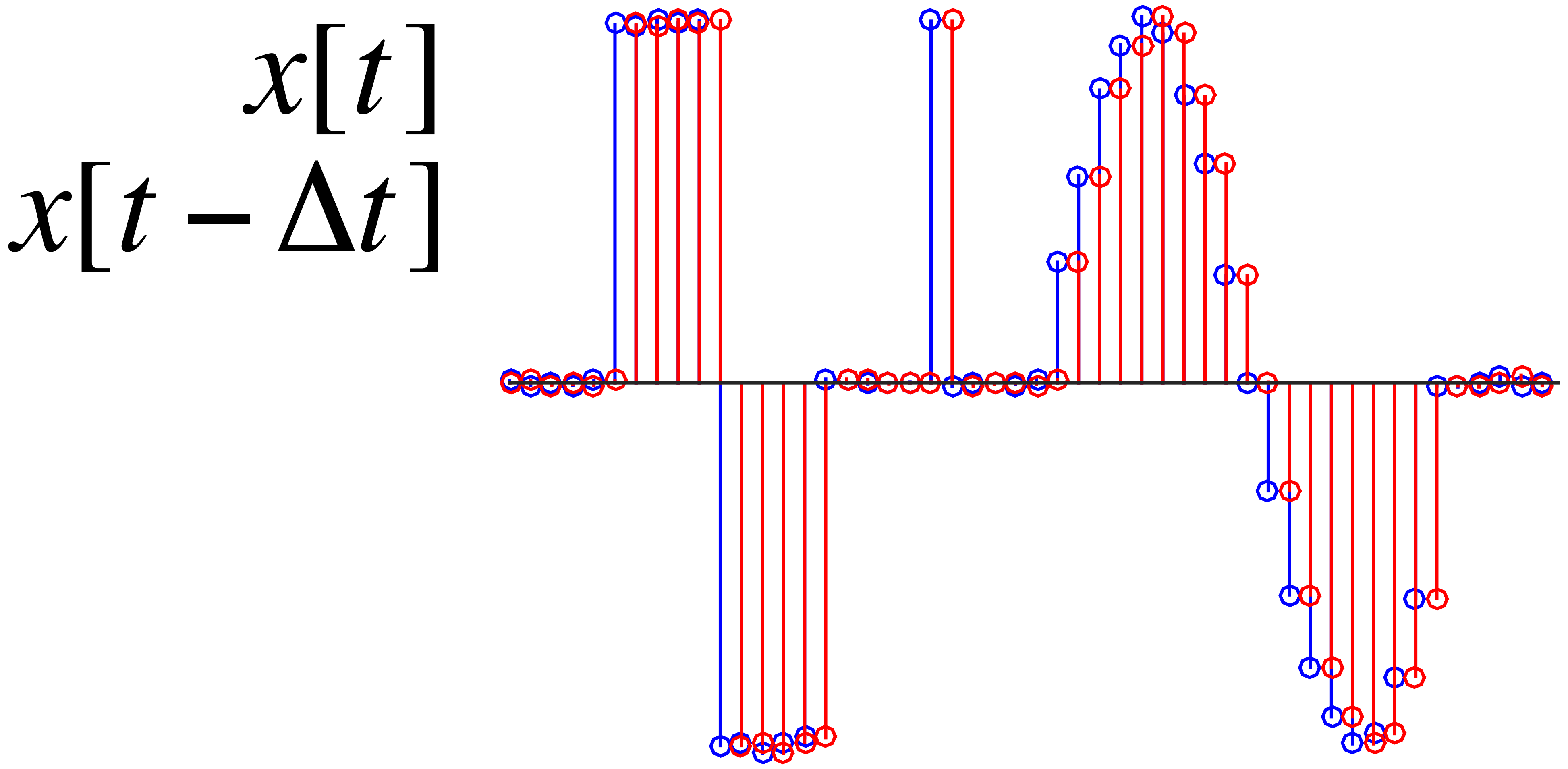


Example: Two-Point Moving Difference



Example: Two-Point Moving Difference

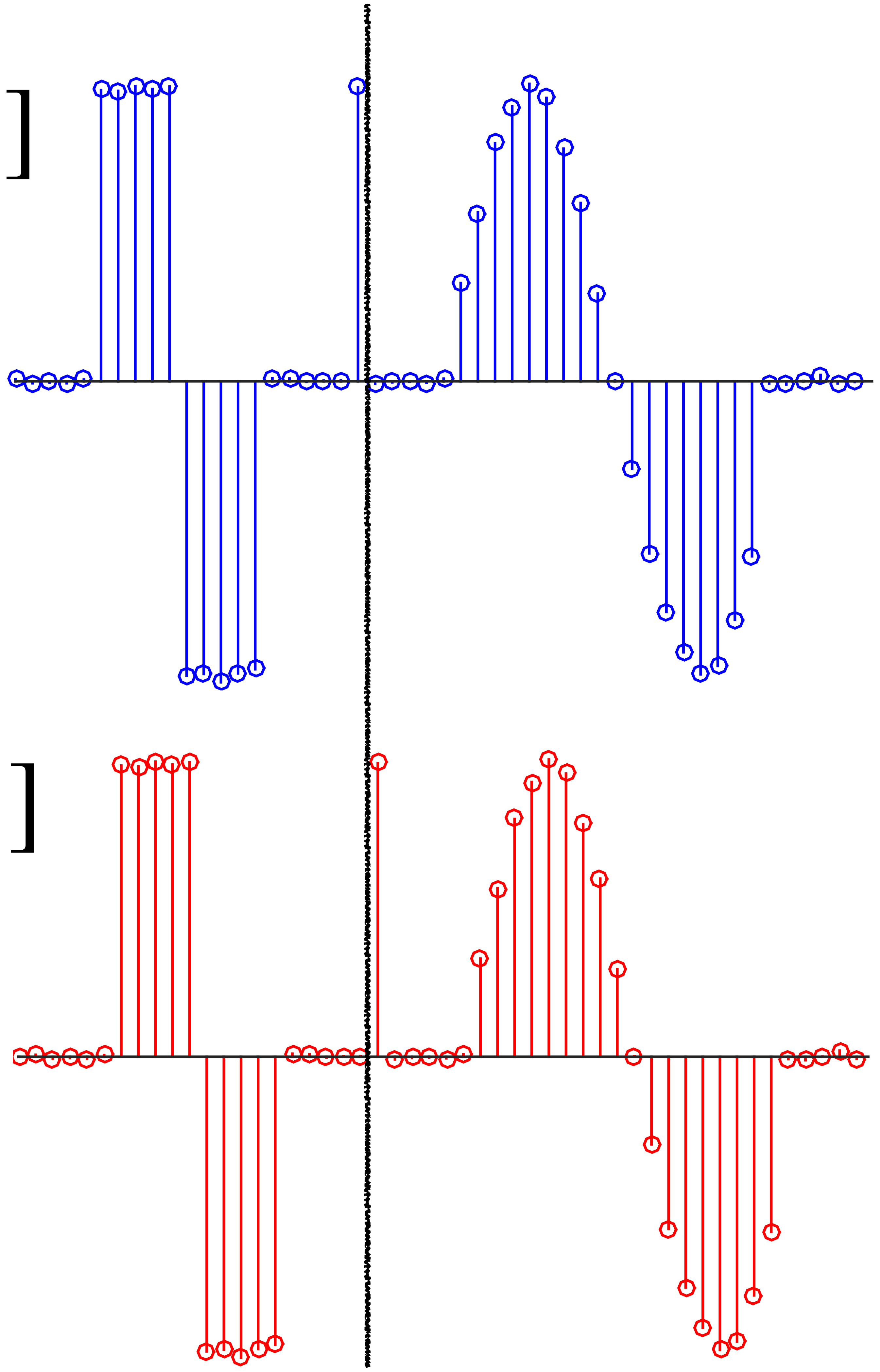




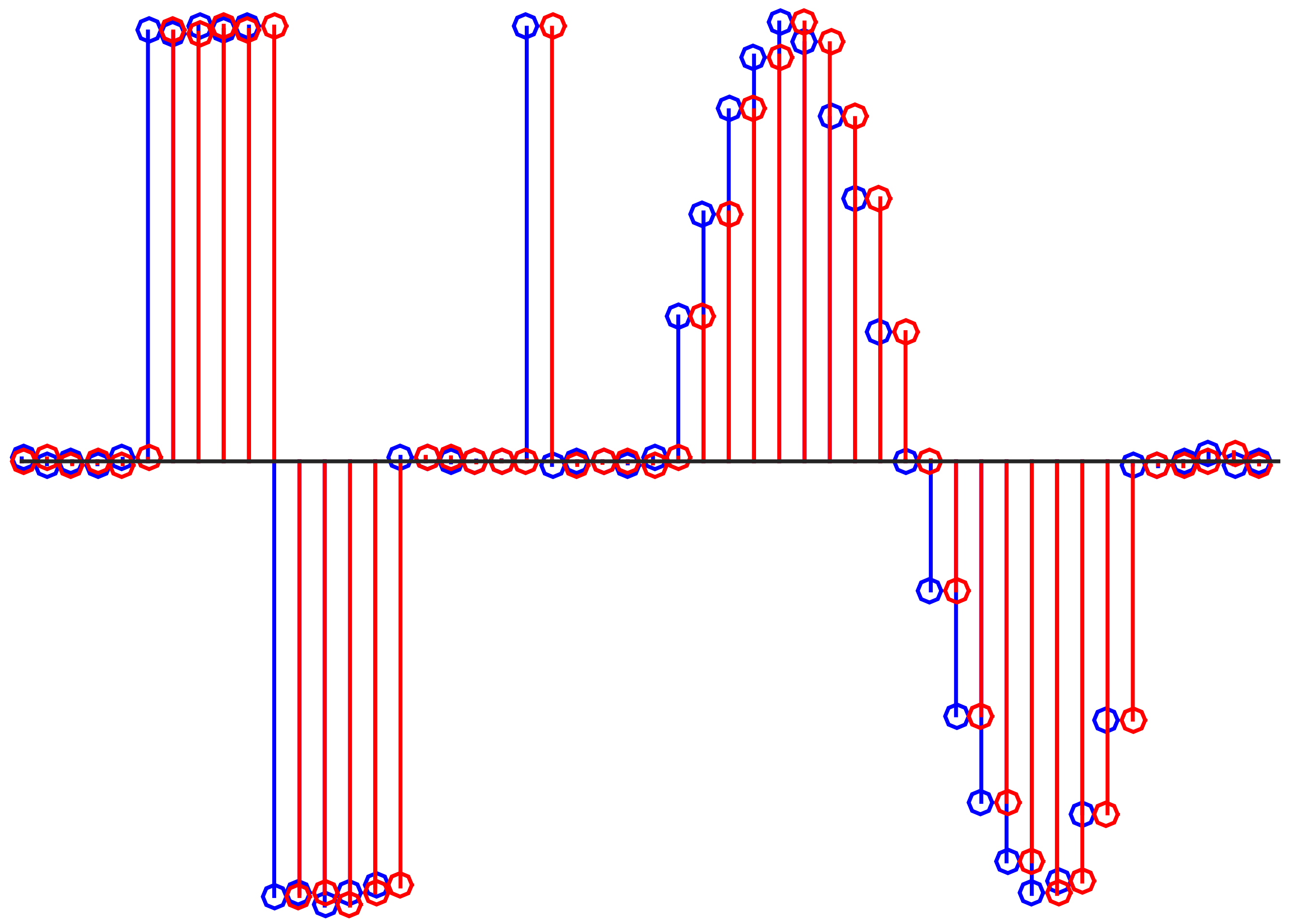
Example: Two-Point Moving Difference

$x[t]$

$x[t-\Delta t]$



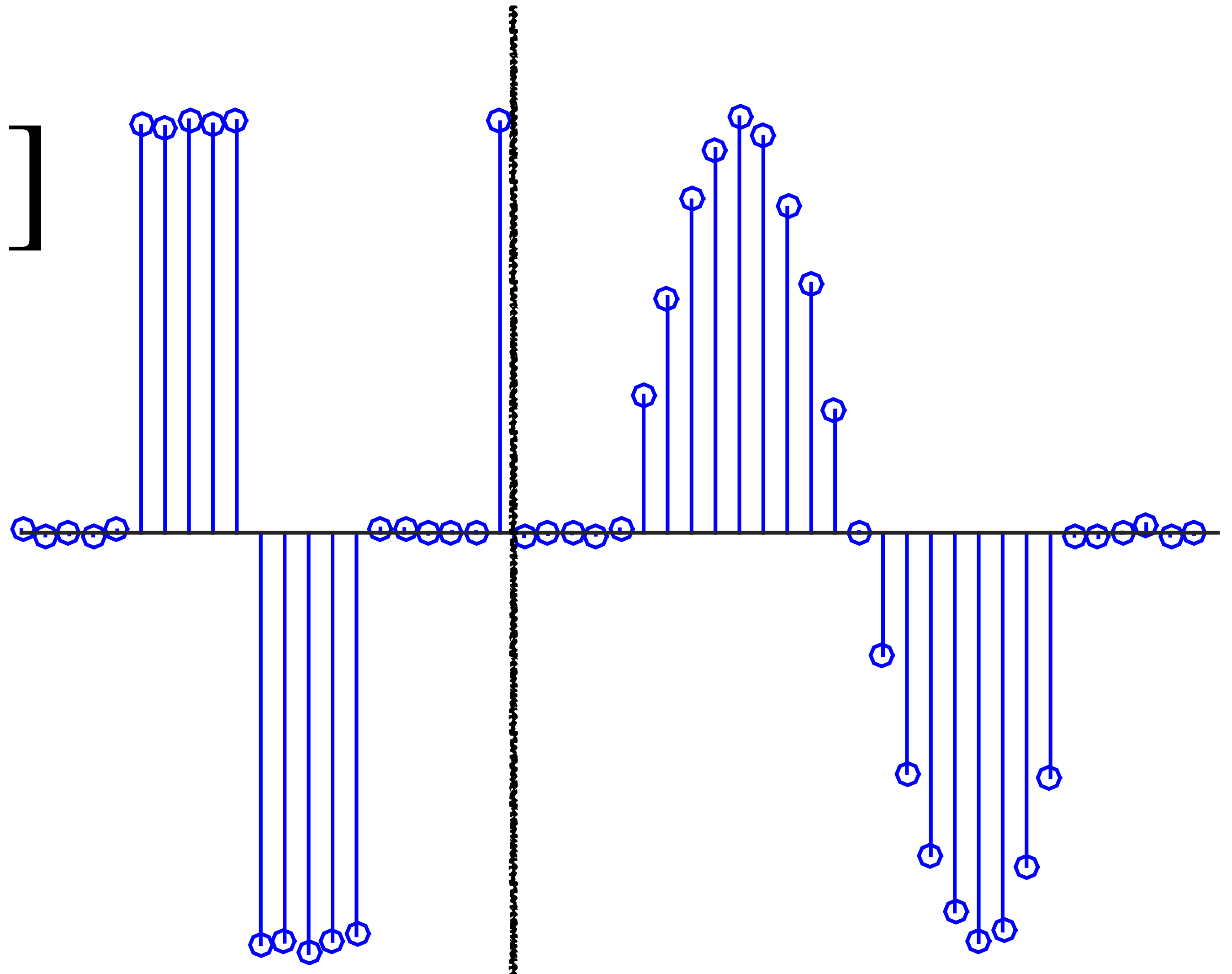
$x[t]$
$x[t-\Delta t]$



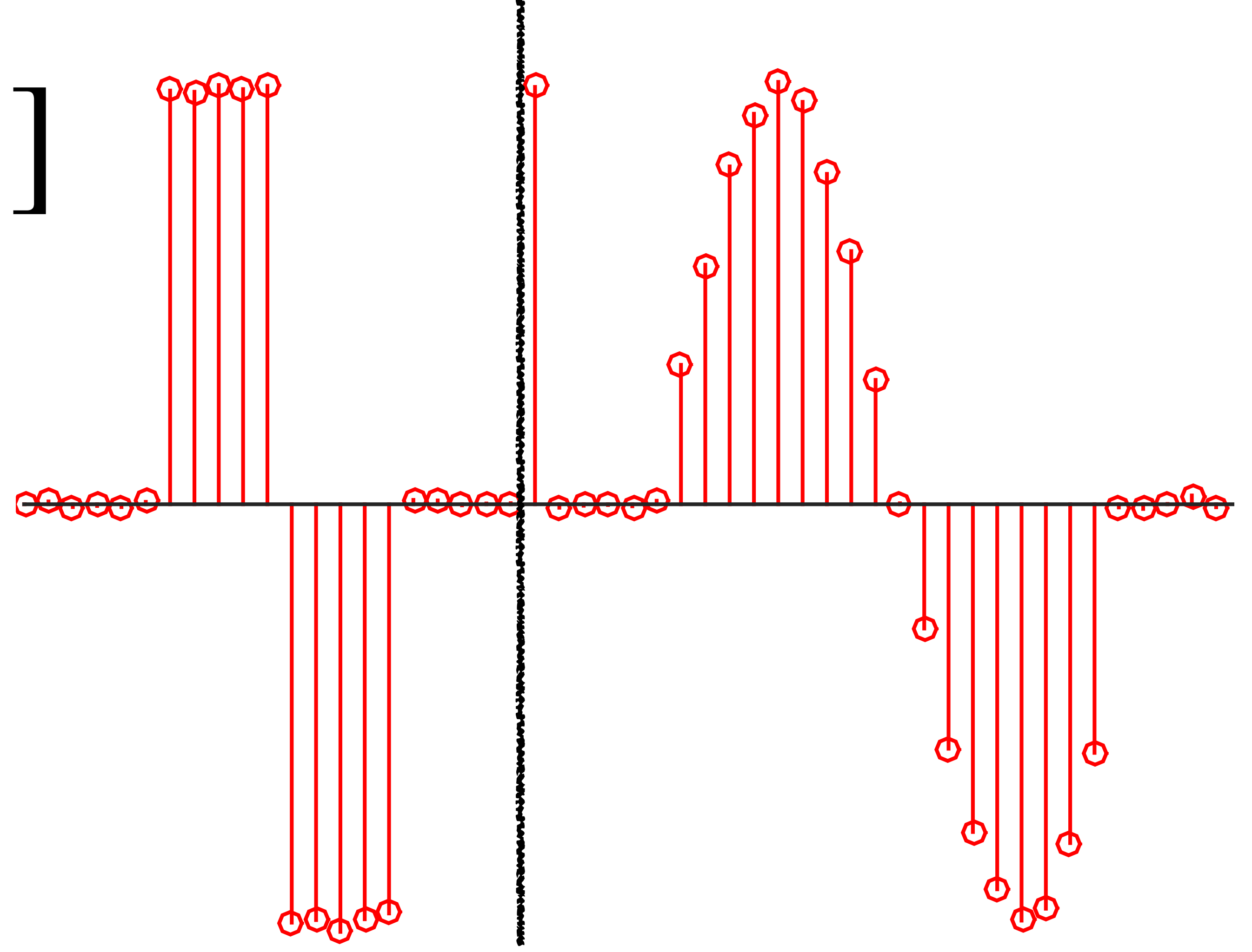
$$\begin{array}{r} x[t]/2 \\ -\ \underline{x[t-\Delta t]/2} \end{array}$$

Example: Two-Point Moving Difference

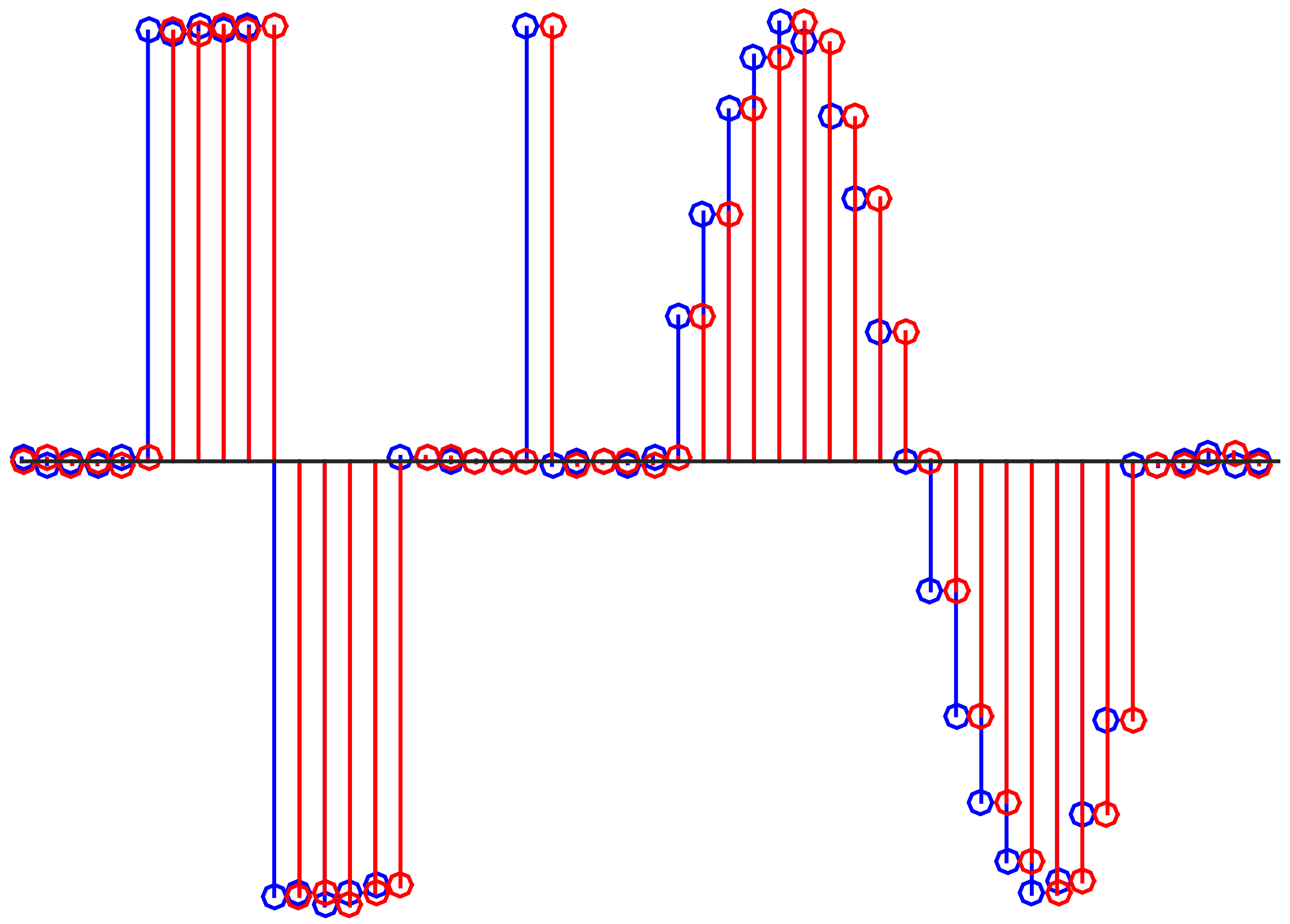
$x[t]$



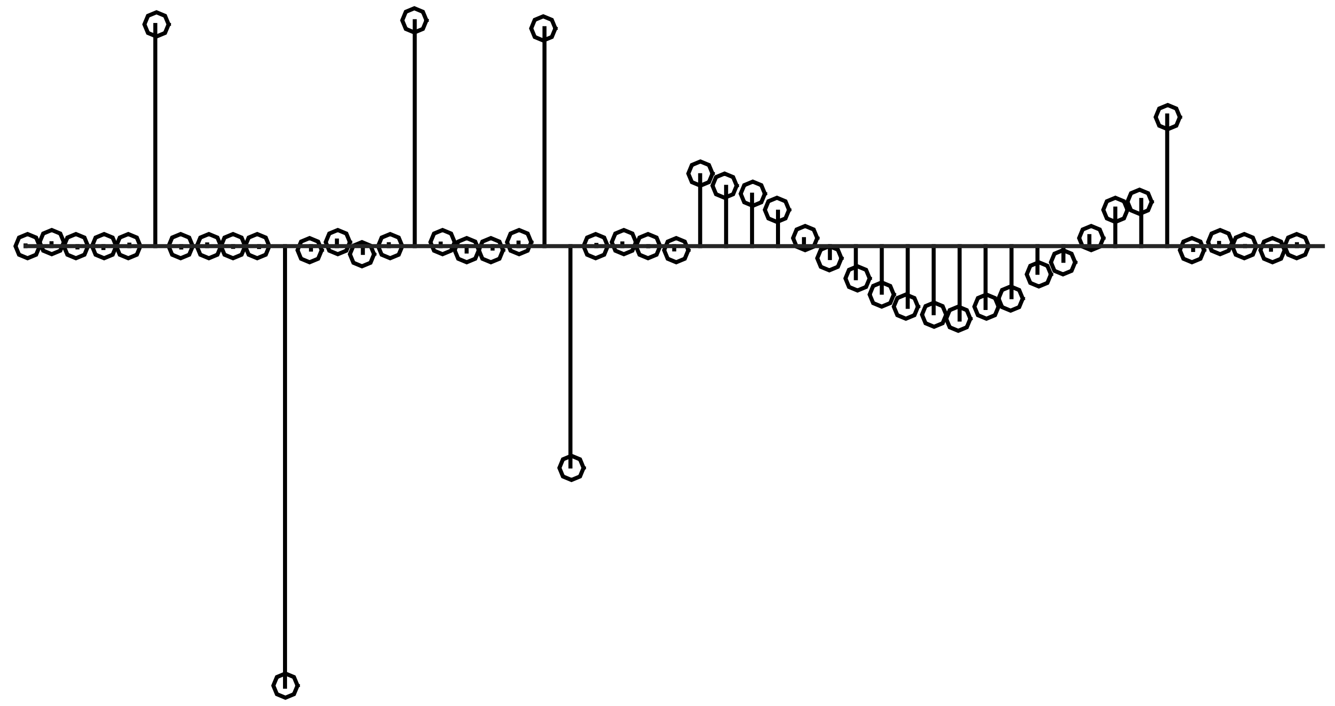
$x[t-\Delta t]$



$x[t]$
$x[t-\Delta t]$

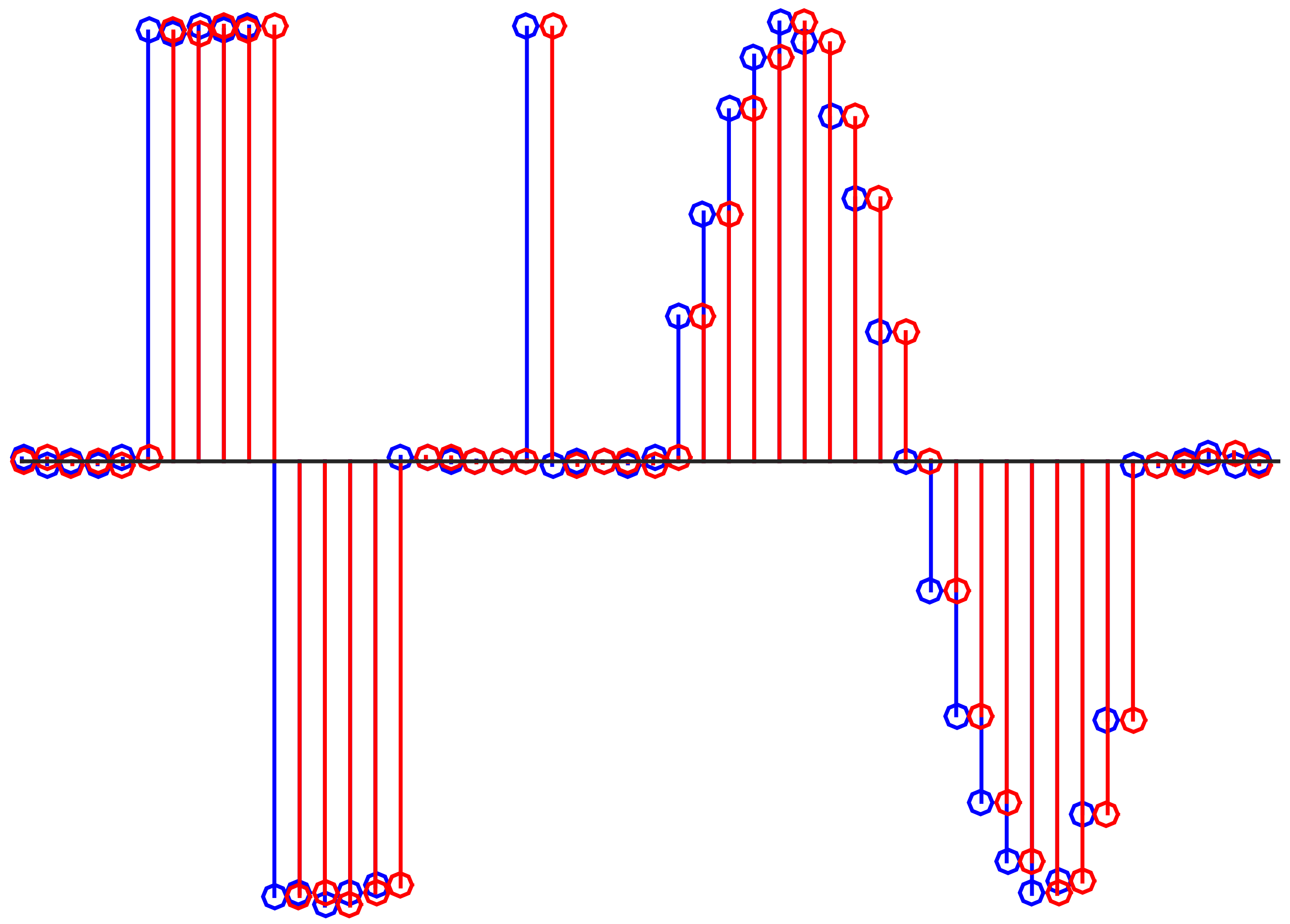


$$\frac{x[t]/2 - x[t-\Delta t]/2}{}$$

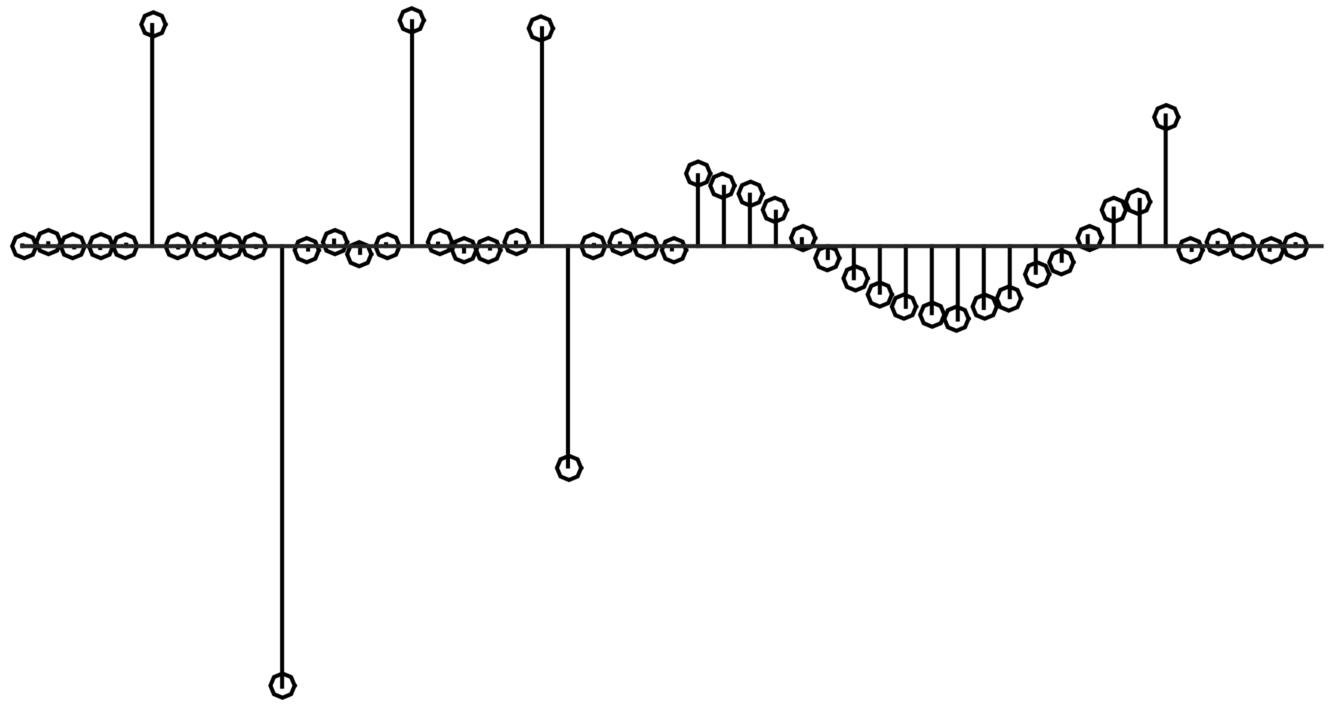


Example: Two-Point Moving Difference

$$x[t]$$
$$x[t-\Delta t]$$

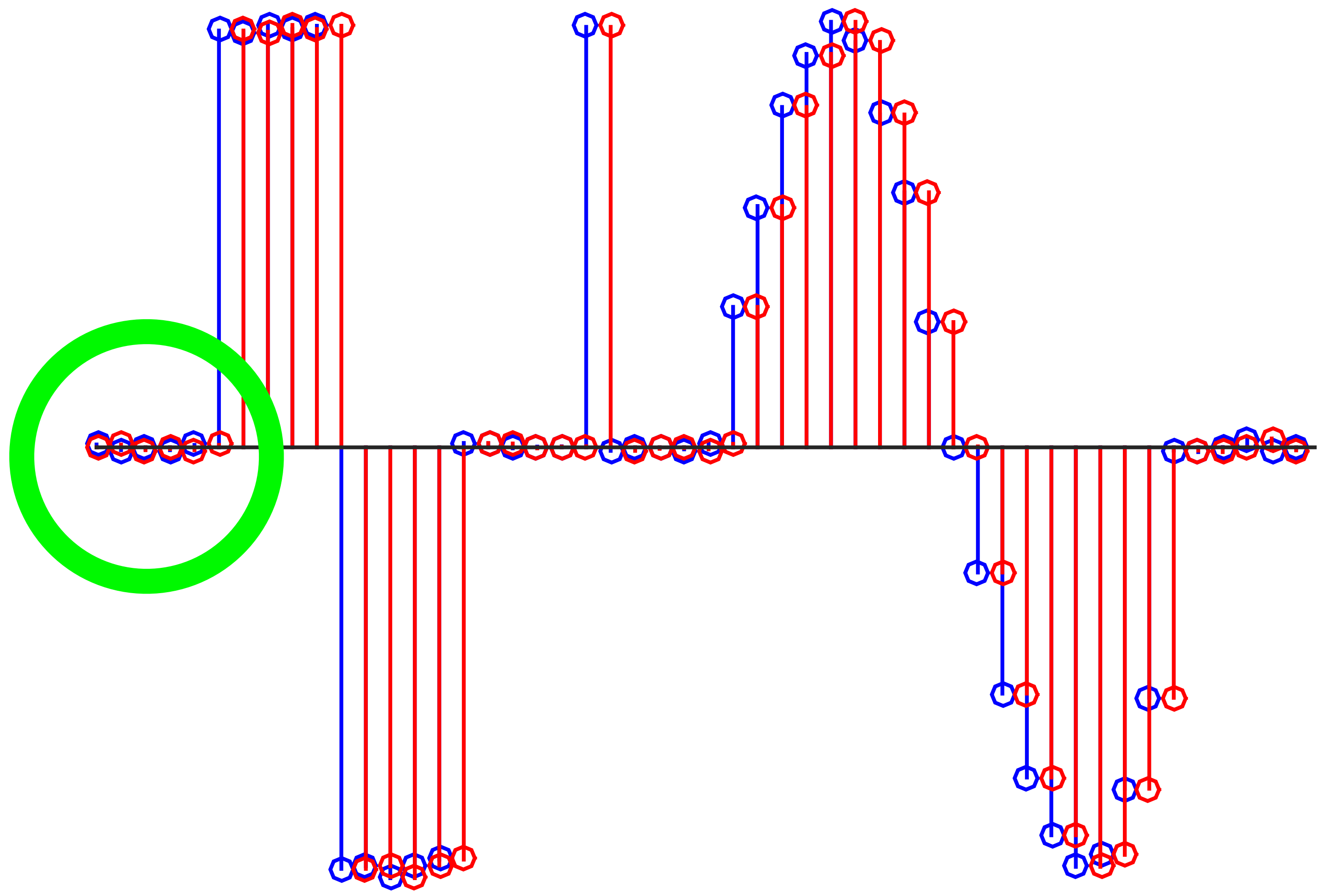


$$y[t] = \frac{1}{2}x[t] - \frac{1}{2}x[t-\Delta t]$$

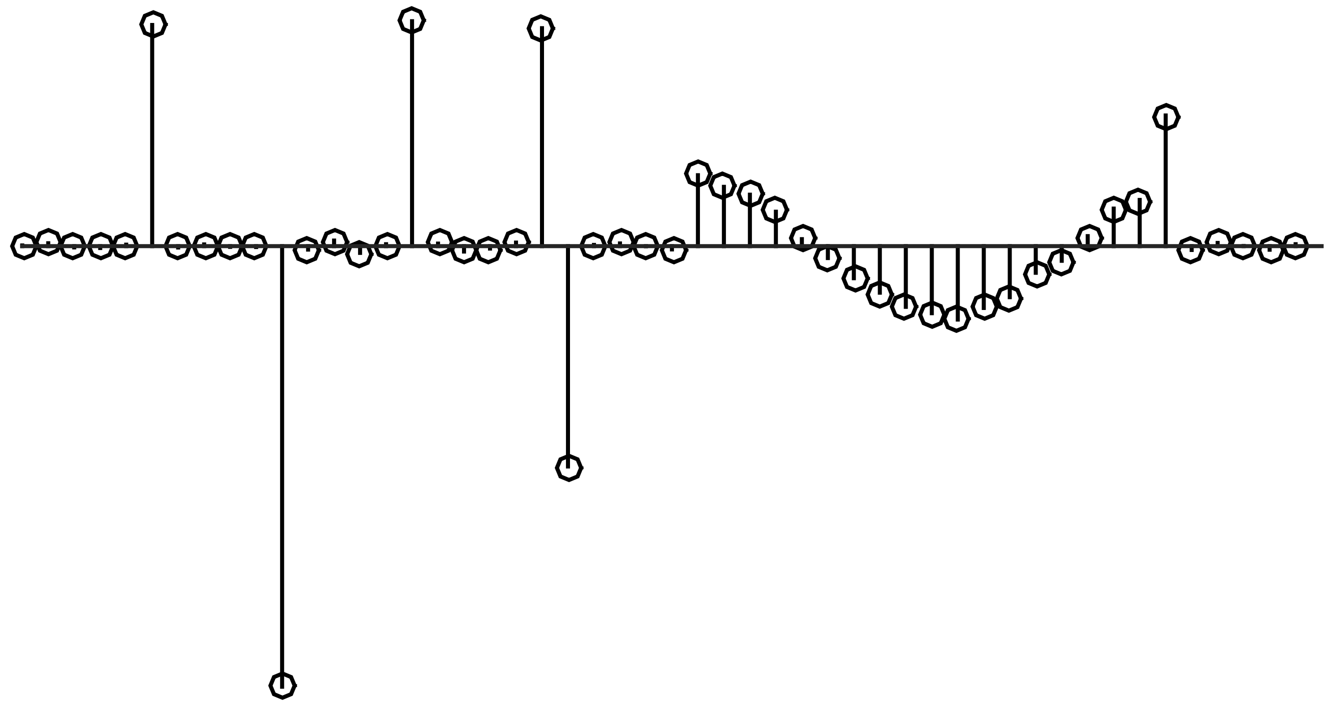


Example: Two-Point Moving Difference

$$x[t]$$
$$x[t-\Delta t]$$

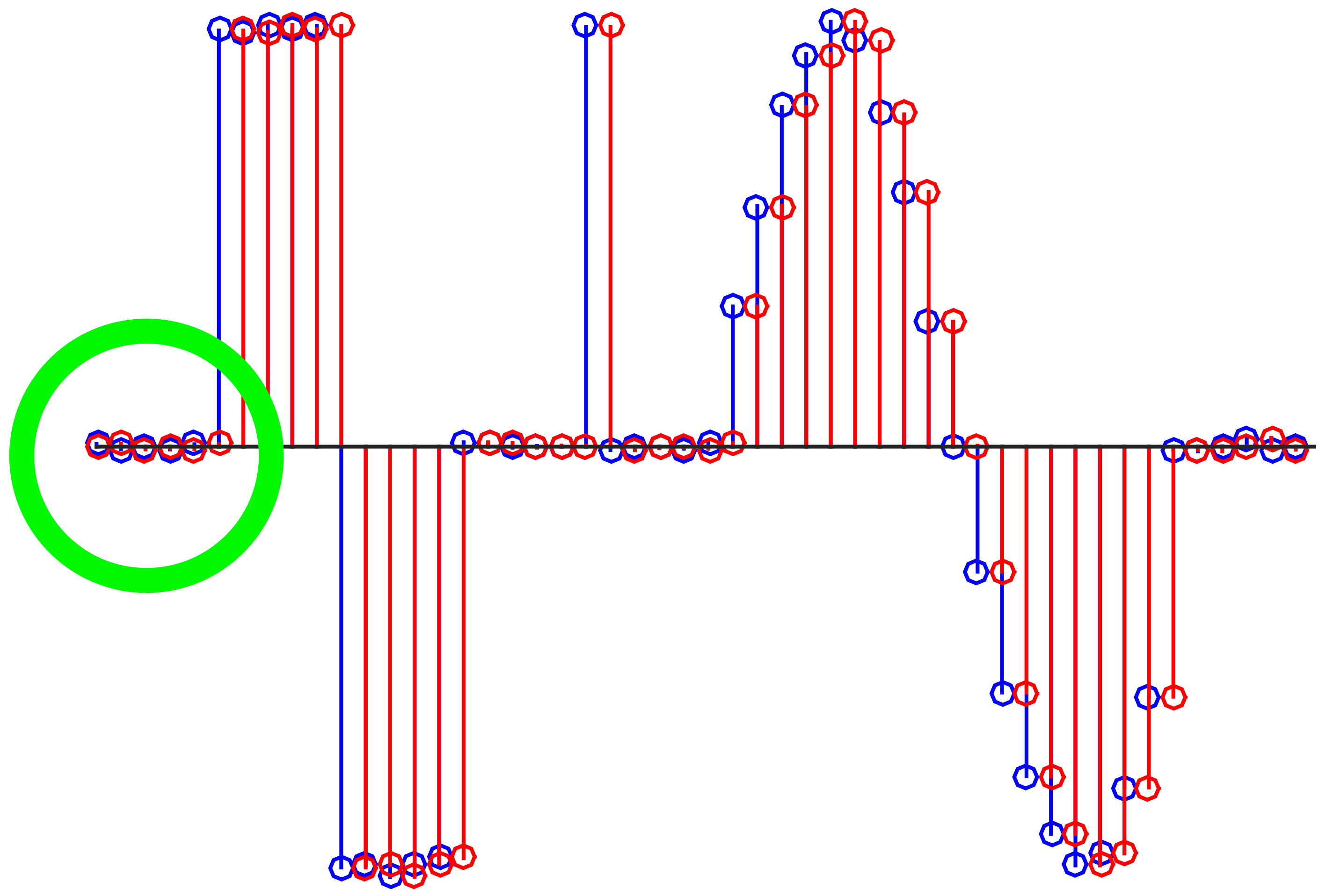


$$y[t]=\frac{1}{2}x[t]-\frac{1}{2}x[t-\Delta t]$$

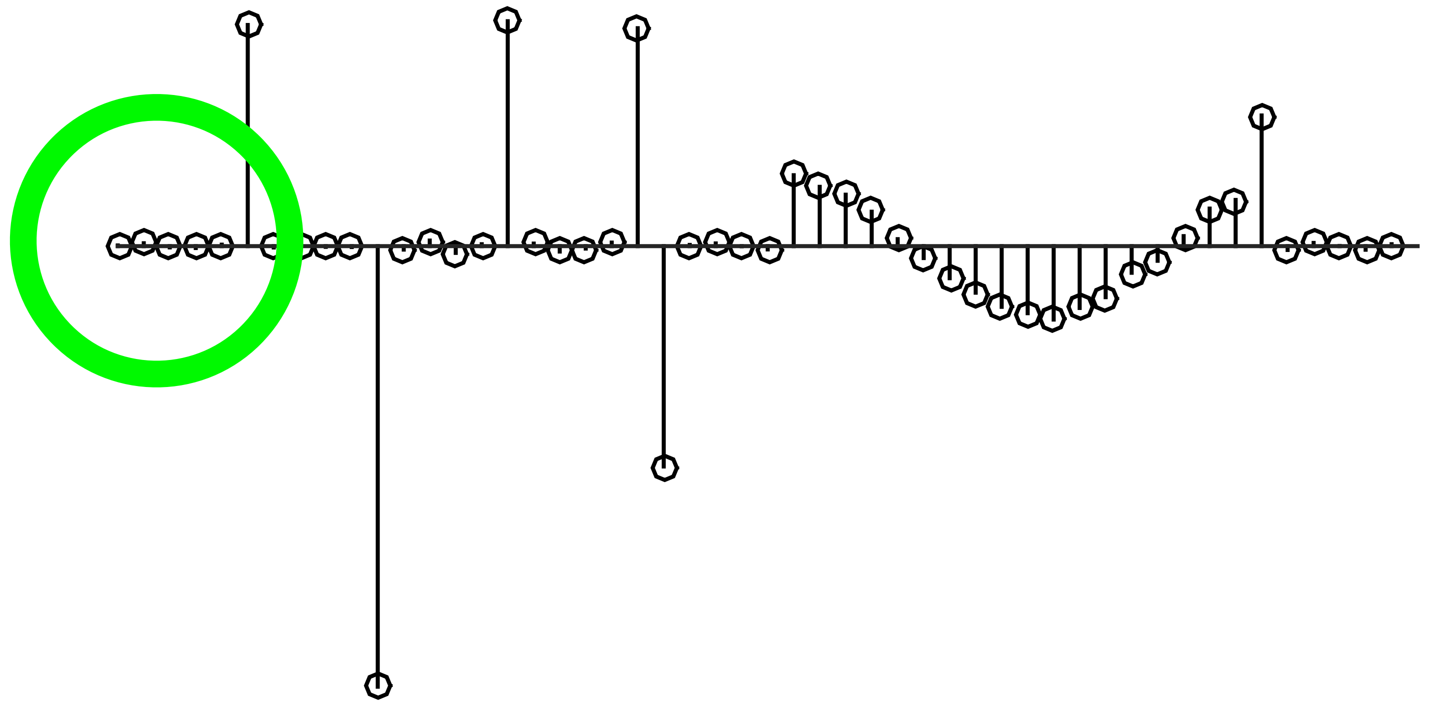


Example: Two-Point Moving Difference

$$x[t]$$
$$x[t-\Delta t]$$

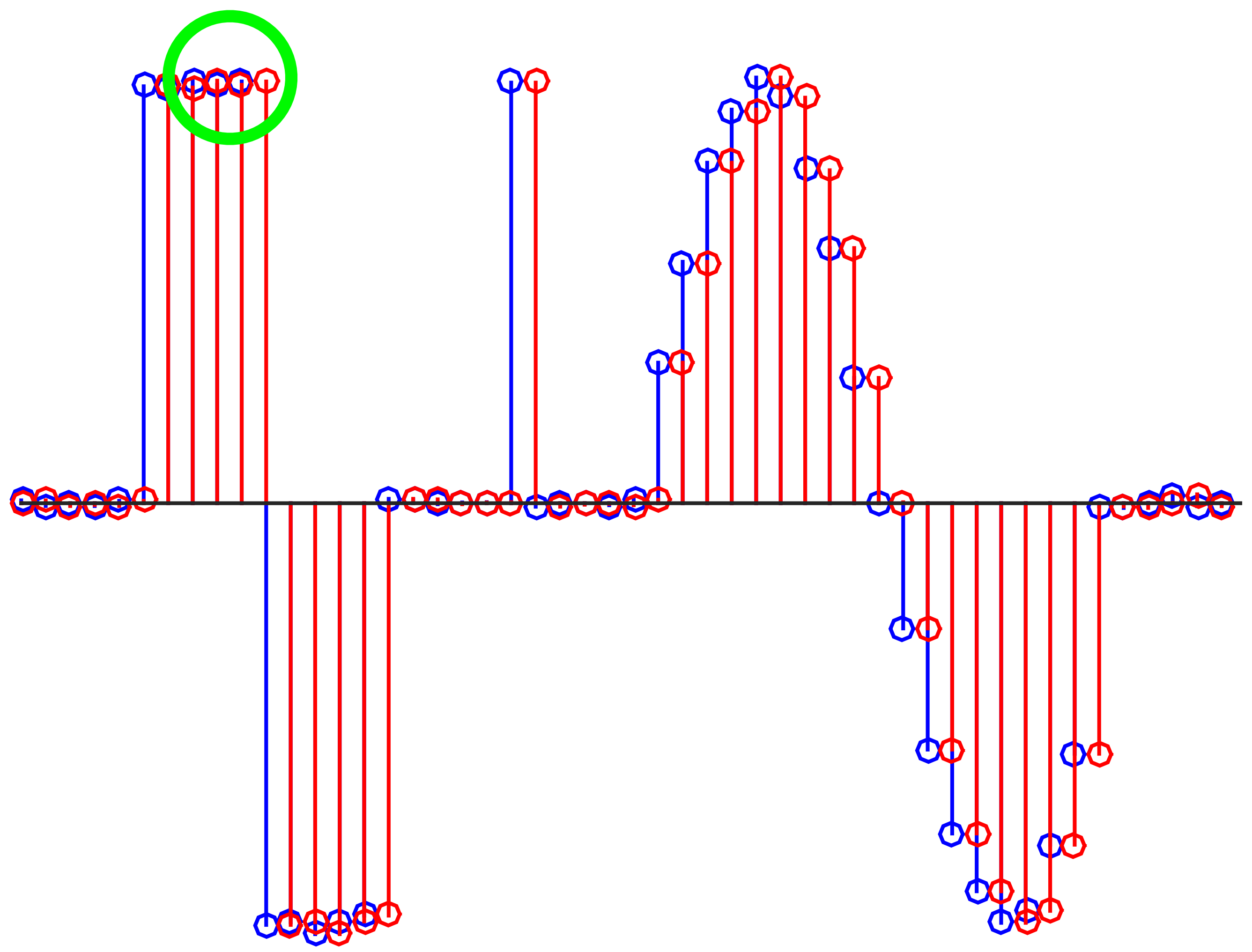


$$y[t] = \frac{1}{2}x[t] - \frac{1}{2}x[t-\Delta t]$$

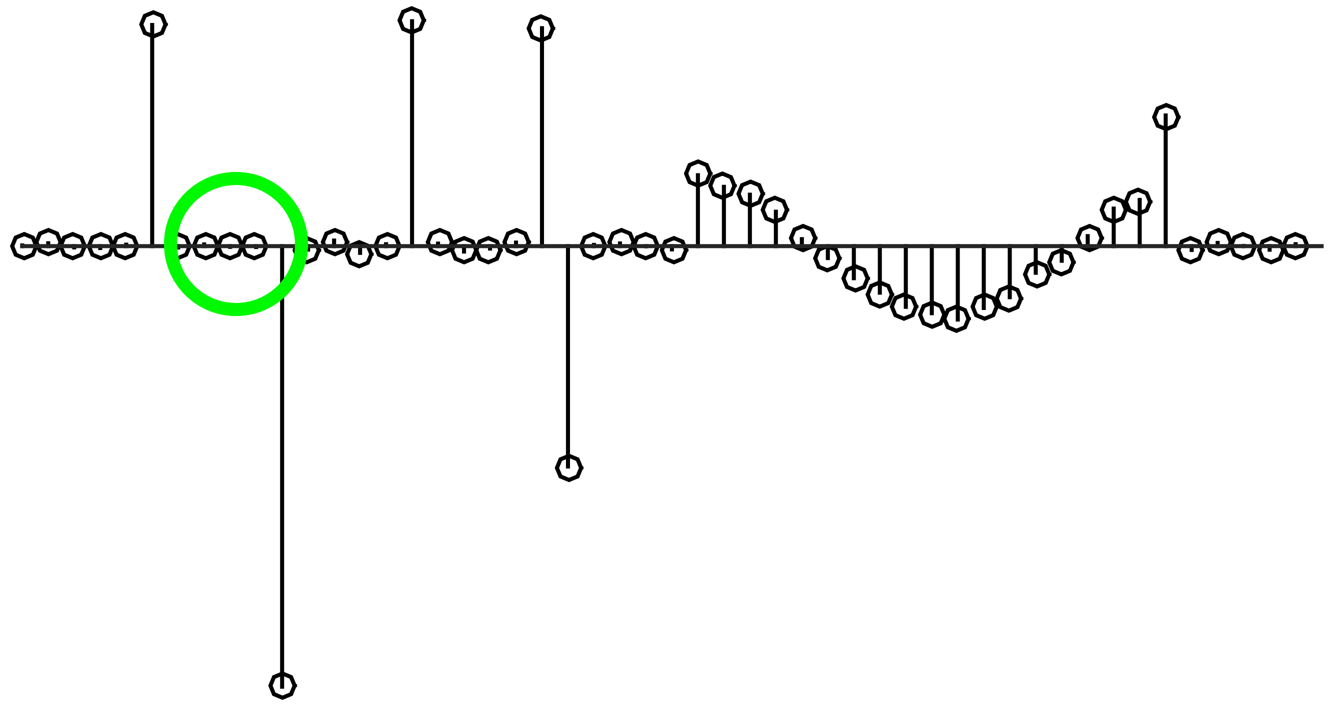


Example: Two-Point Moving Difference

$$x[t]$$
$$x[t-\Delta t]$$

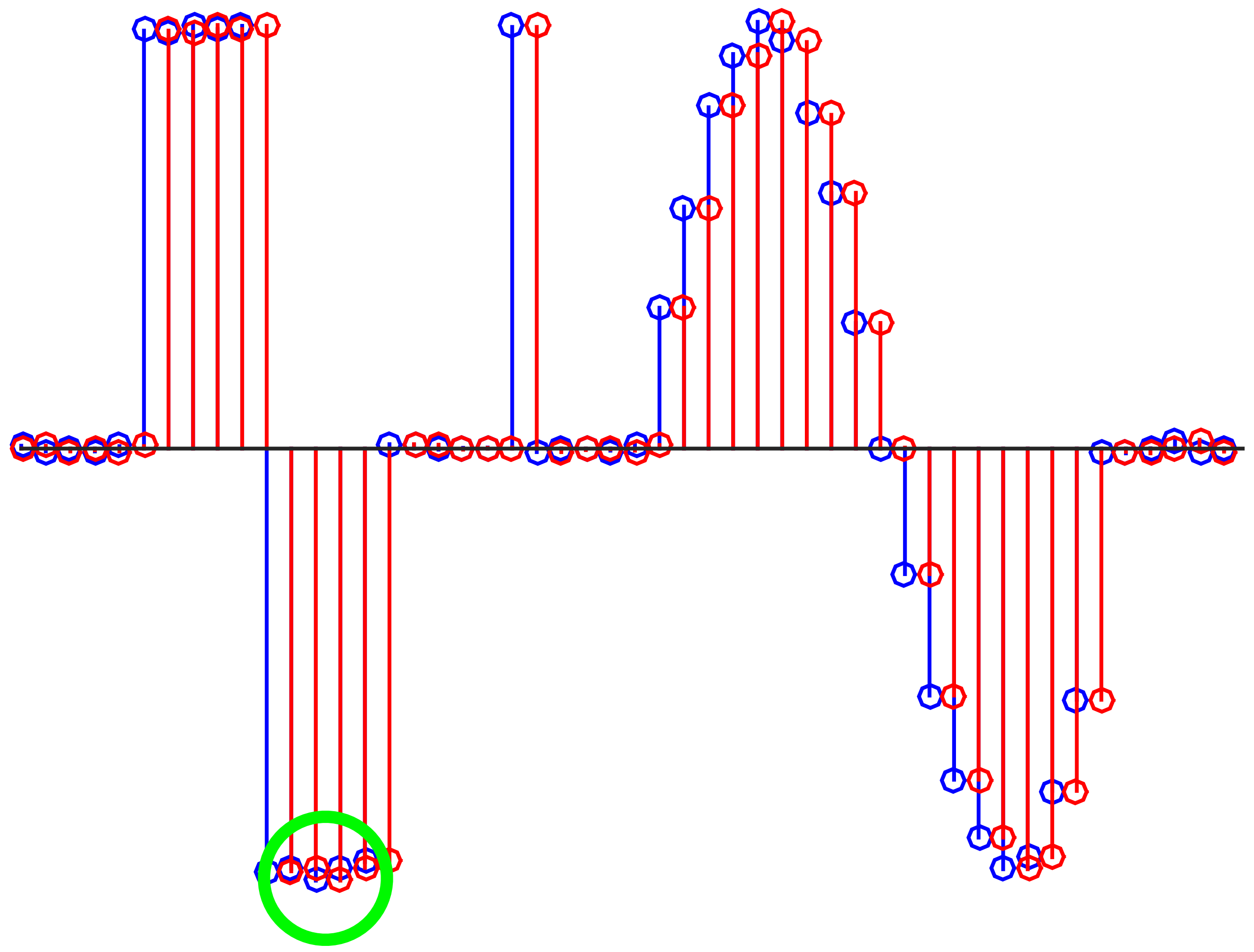


$$y[t] = \frac{1}{2}x[t] - \frac{1}{2}x[t-\Delta t]$$

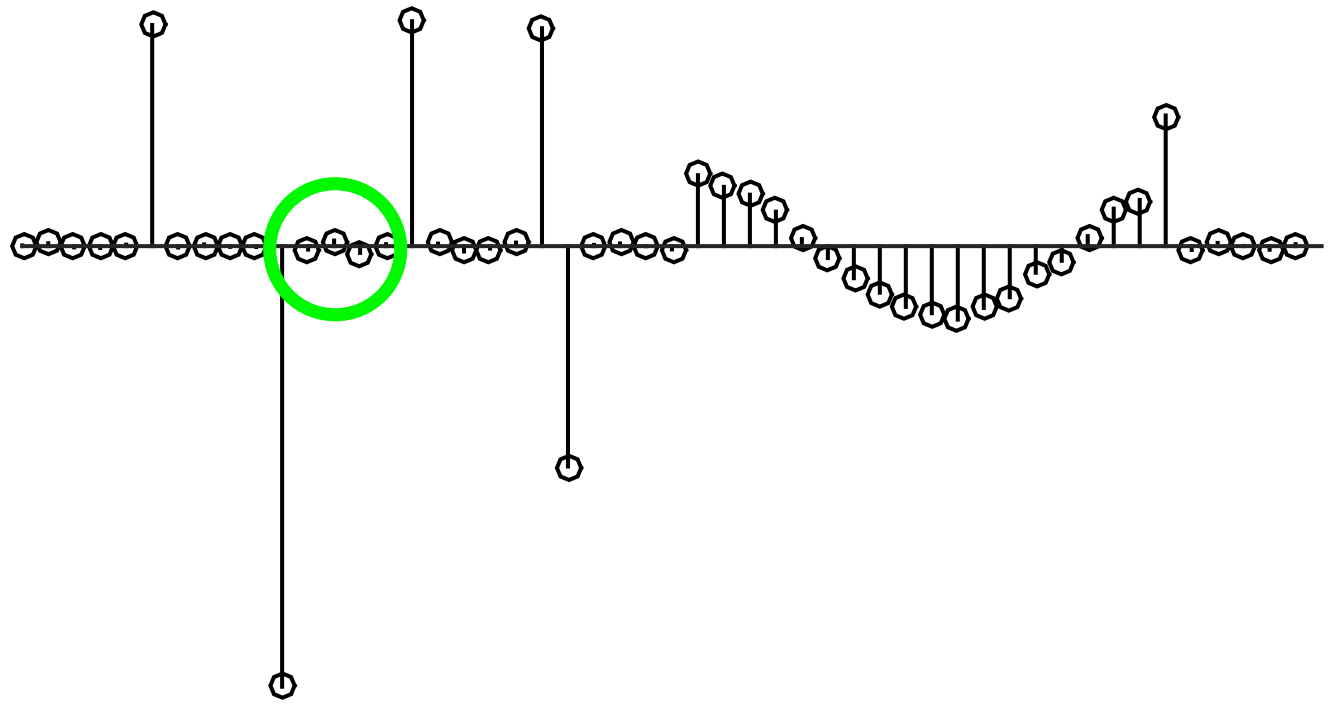


Example: Two-Point Moving Difference

$$x[t]$$
$$x[t-\Delta t]$$

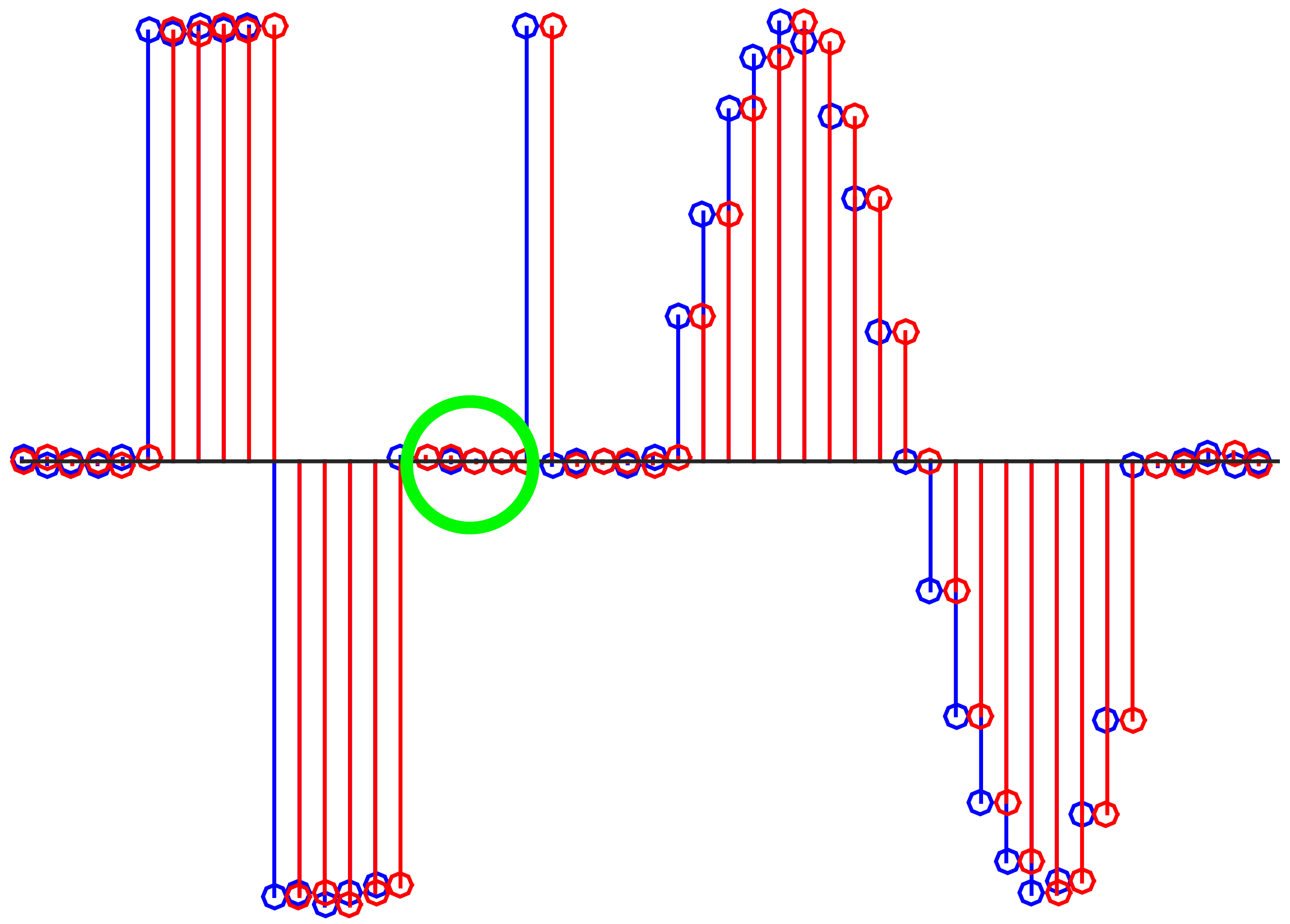


$$y[t] = \frac{1}{2}x[t] - \frac{1}{2}x[t-\Delta t]$$

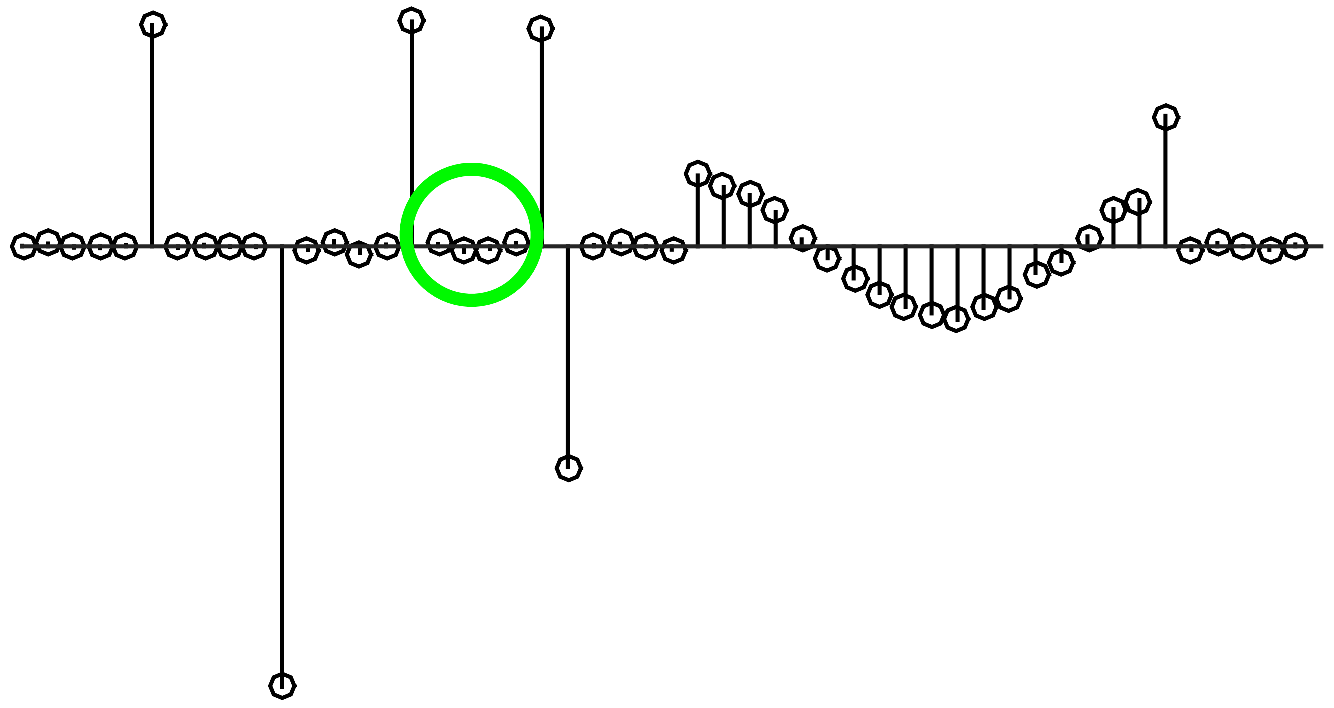


Example: Two-Point Moving Difference

$x[t]$
$x[t-\Delta t]$

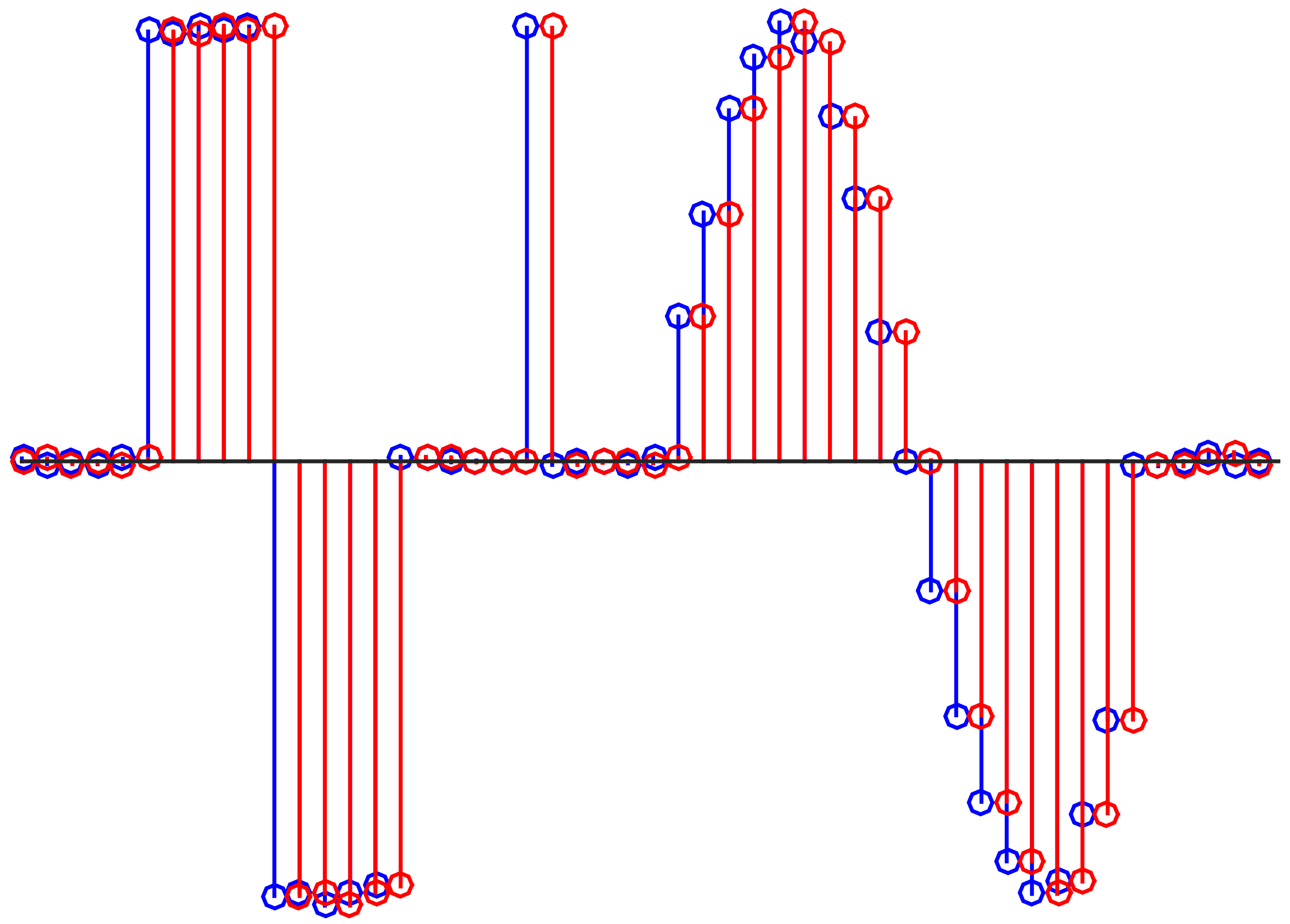


$$y[t]=\frac{1}{2}x[t]-\frac{1}{2}x[t-\Delta t]$$

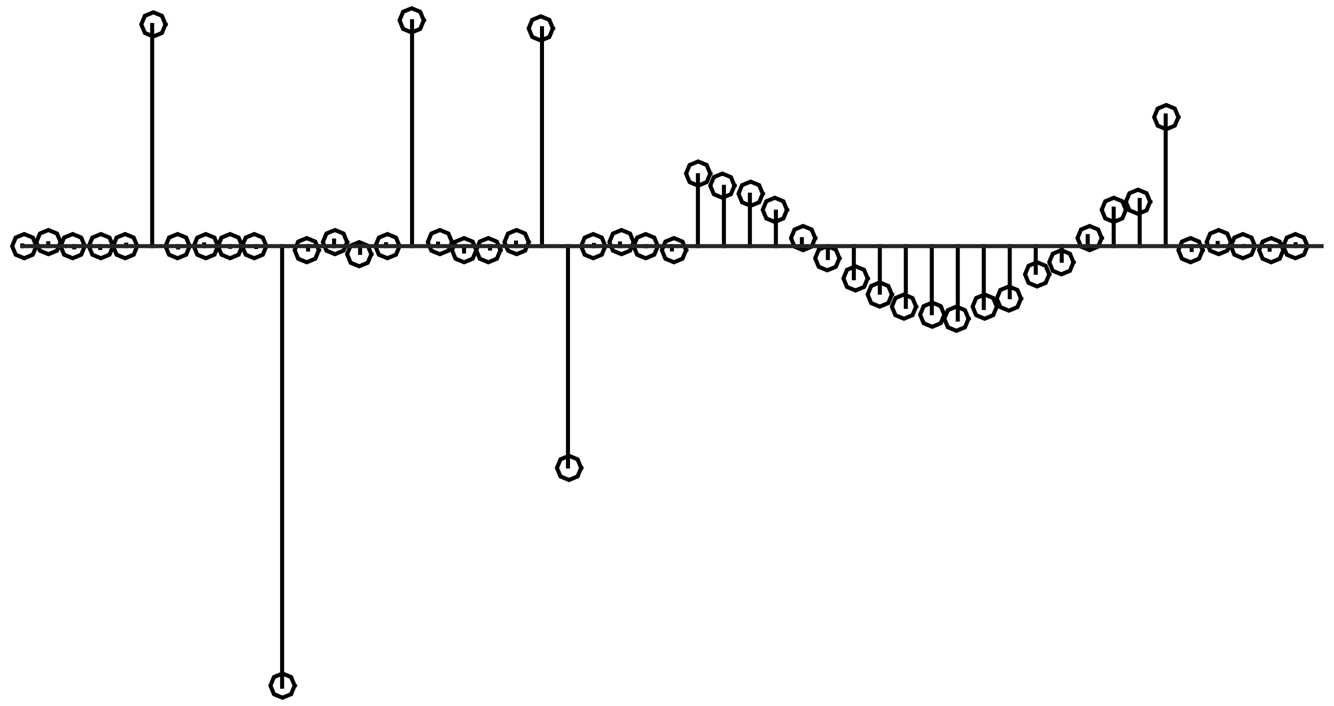


Example: Two-Point Moving Difference

$x[t]$
$x[t-\Delta t]$

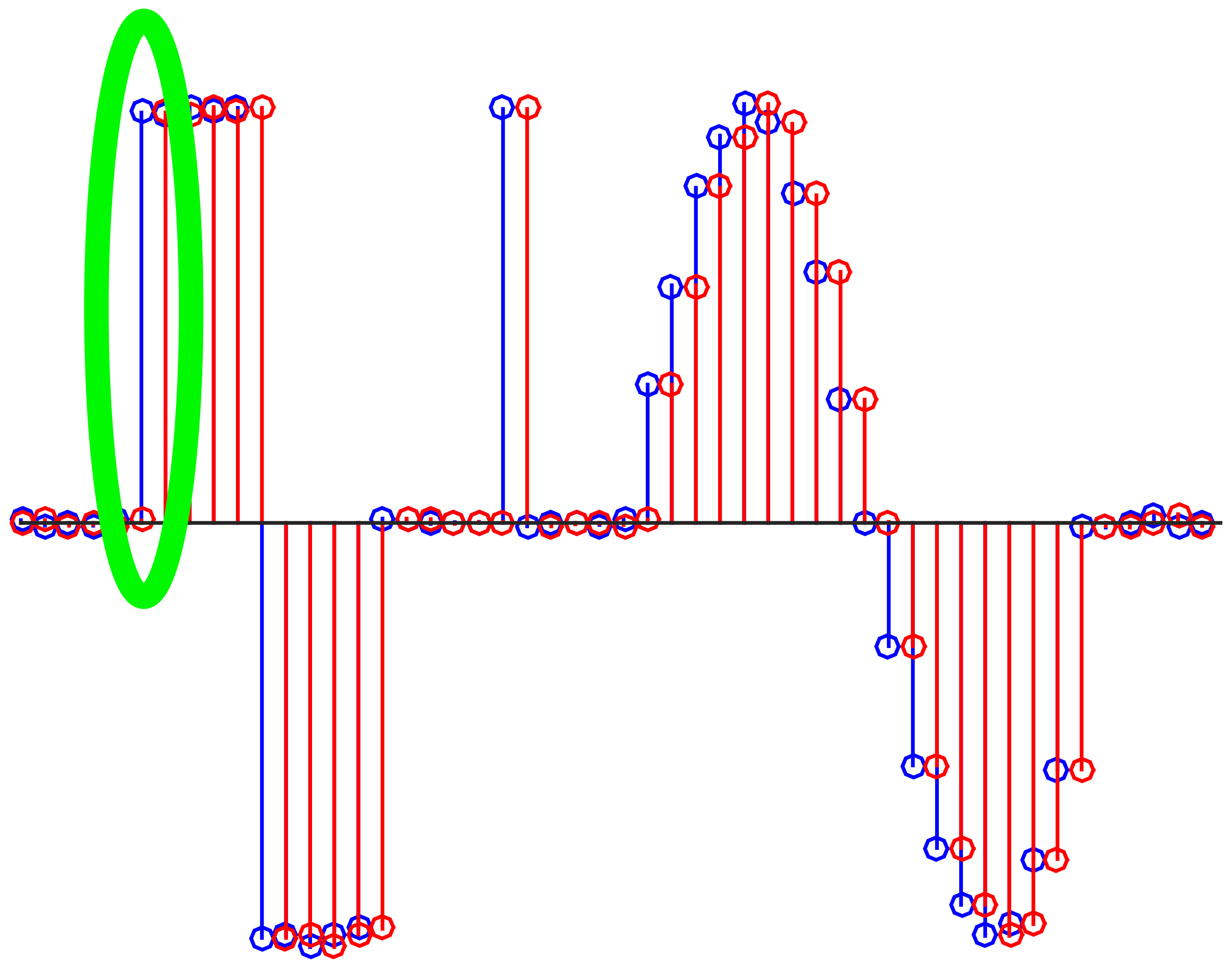


$$y[t] = \frac{1}{2}x[t] - \frac{1}{2}x[t-\Delta t]$$

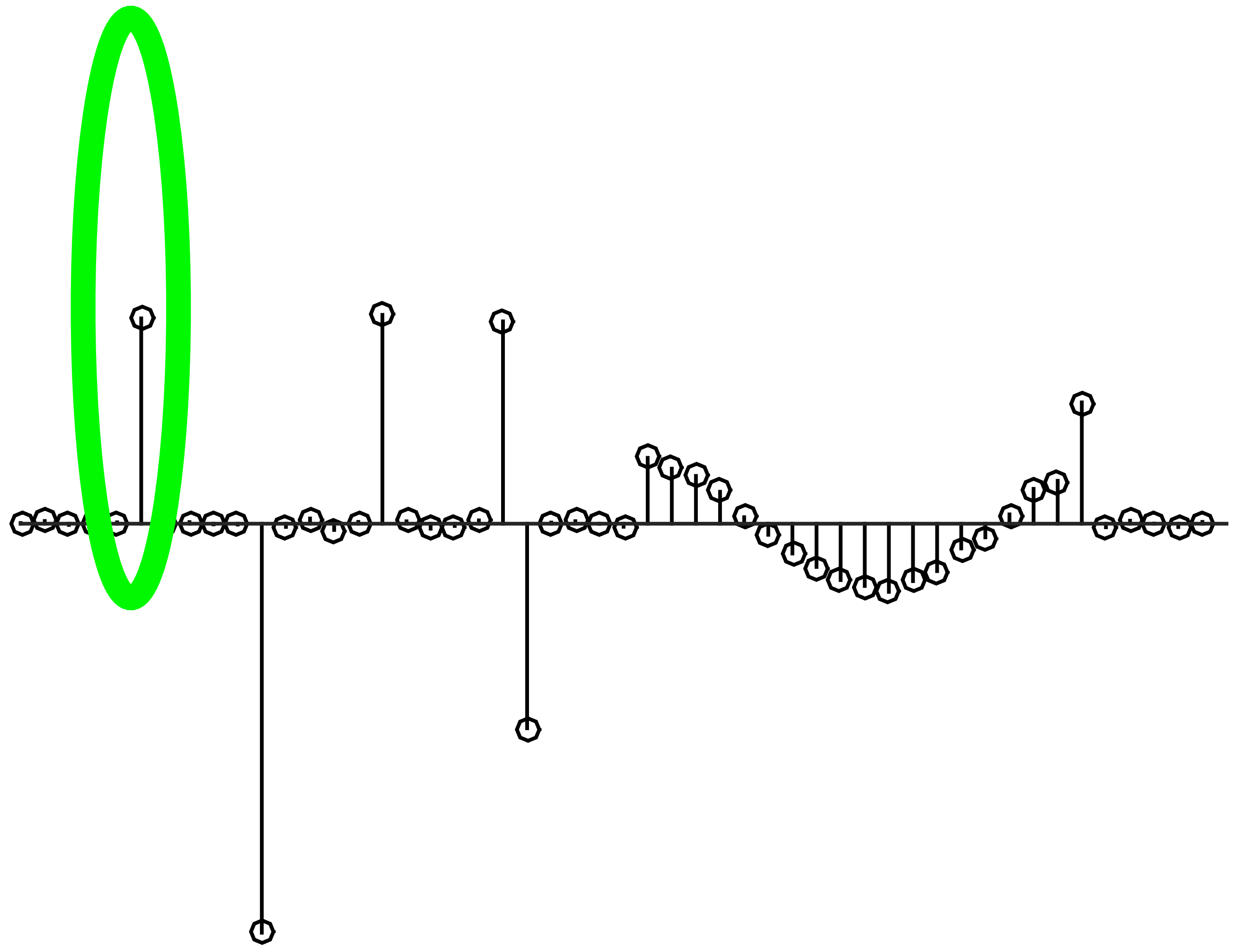


Example: Two-Point Moving Difference

$x[t]$
$x[t-\Delta t]$



$$y[t] = \frac{1}{2}x[t] - \frac{1}{2}x[t-\Delta t]$$

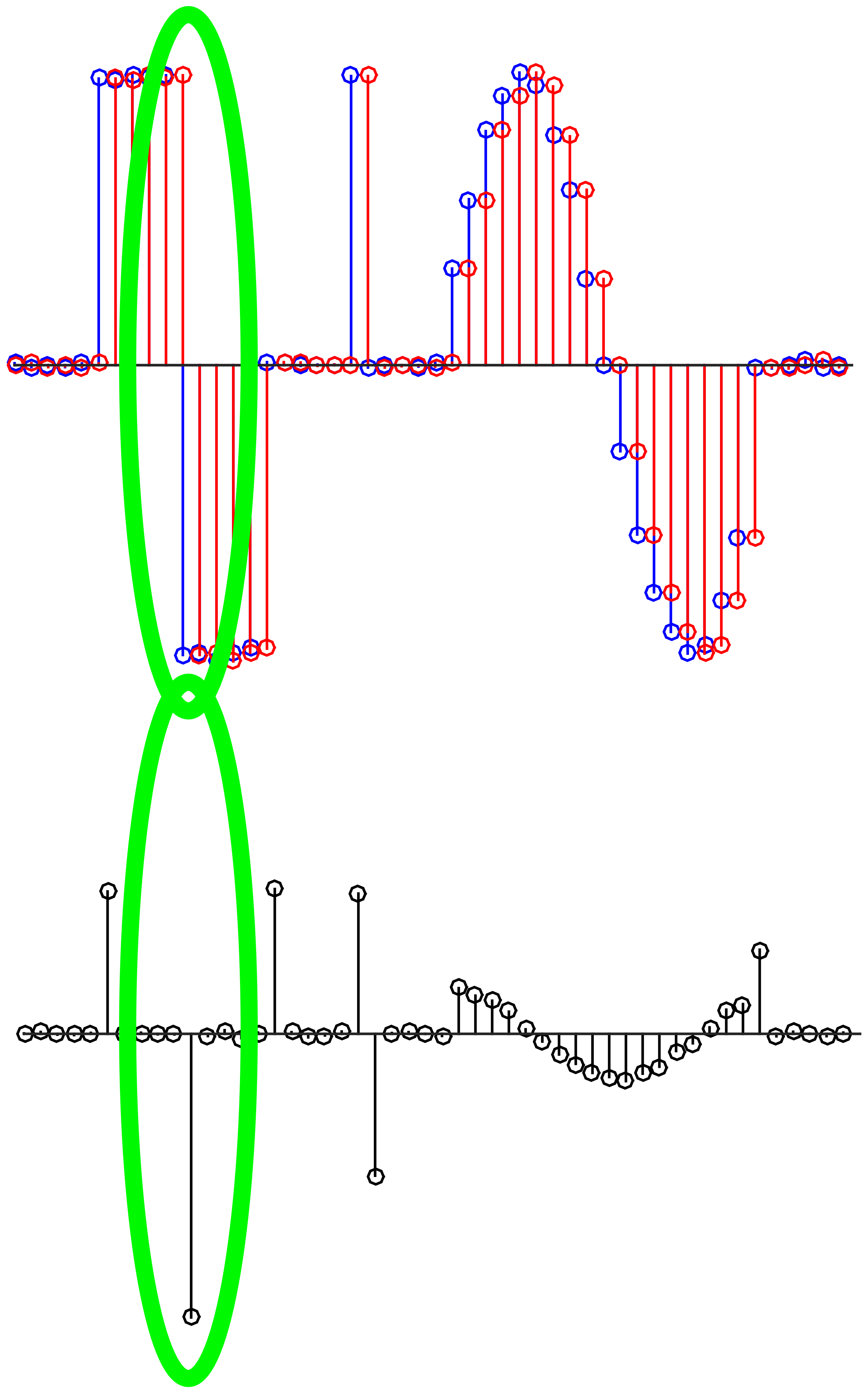


Example: Two-Point Moving Difference

$x[t]$

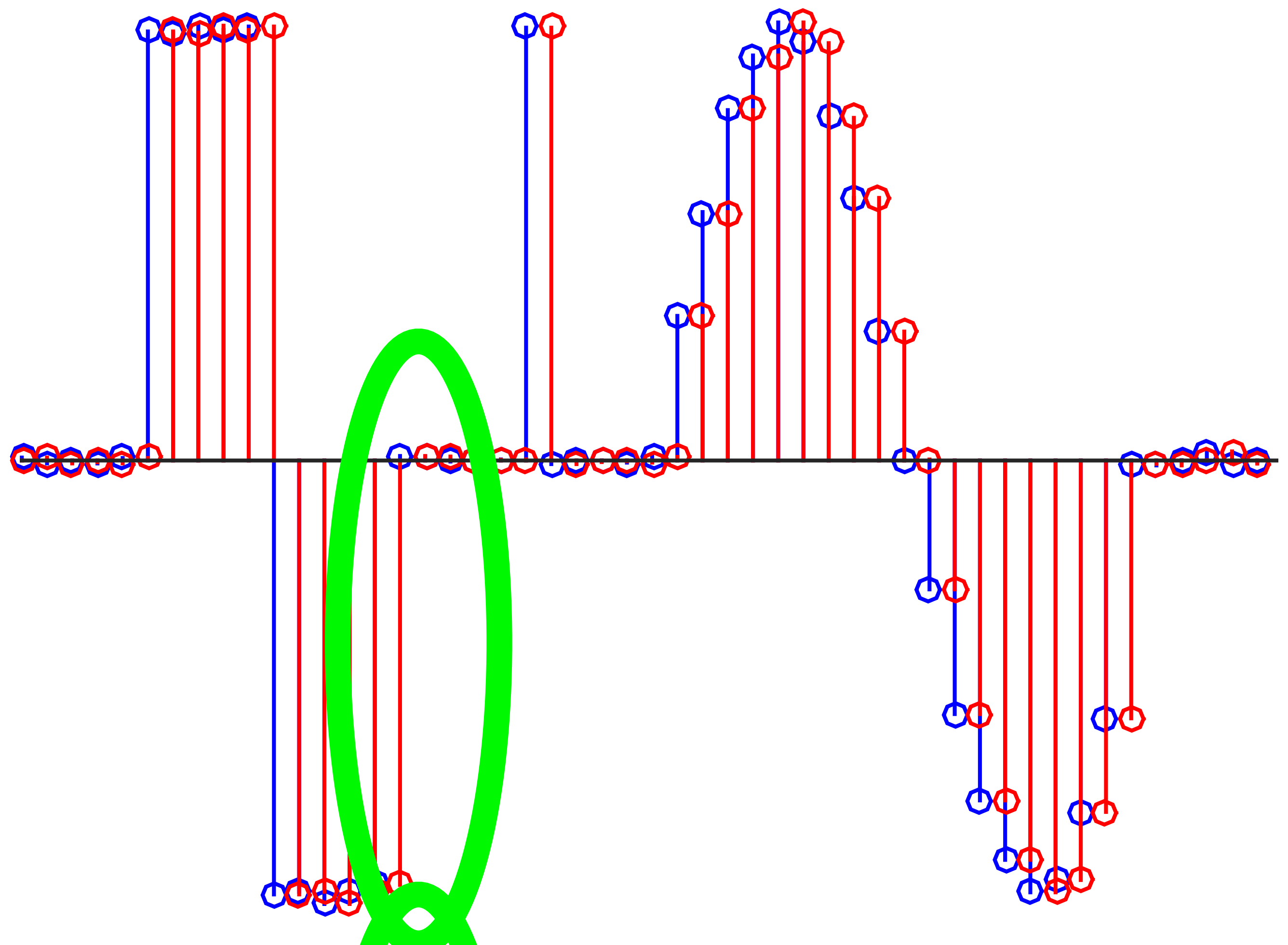
$x[t-\Delta t]$

$$y[t] = \frac{1}{2}x[t] - \frac{1}{2}x[t-\Delta t]$$

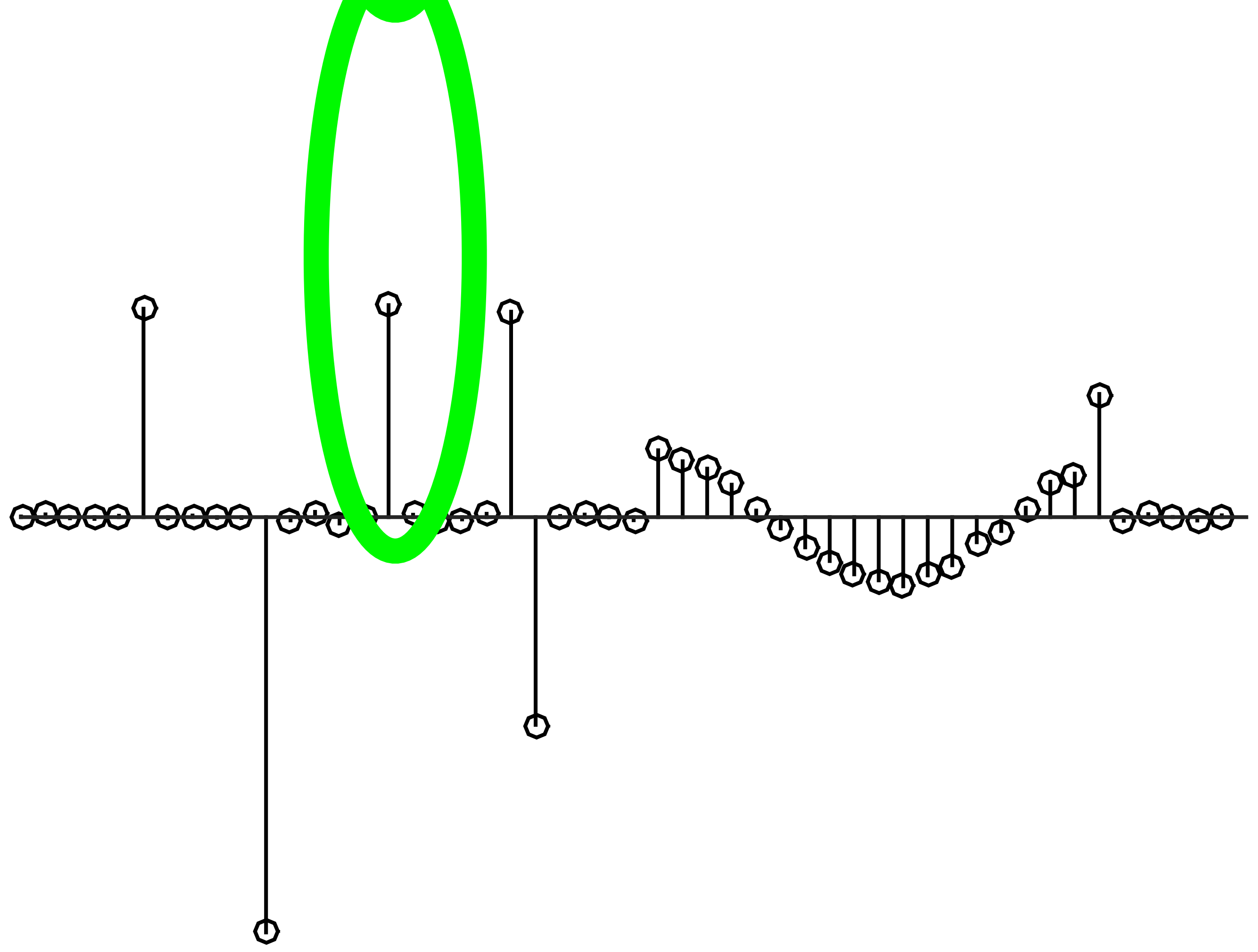


Example: Two-Point Moving Difference

$x[t]$
$x[t - \Delta t]$

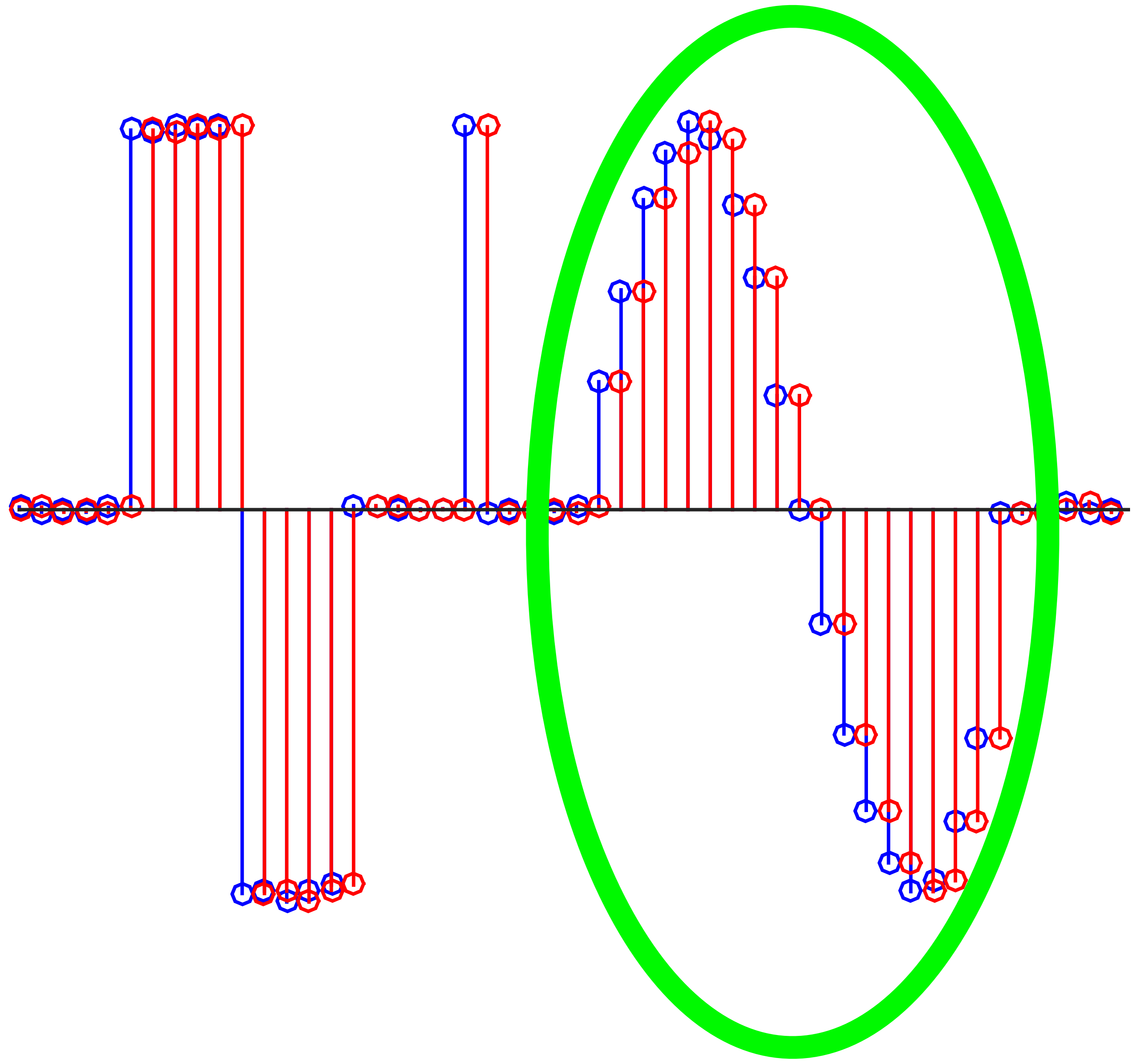


$$y[t] = \frac{1}{2}x[t] - \frac{1}{2}x[t - \Delta t]$$

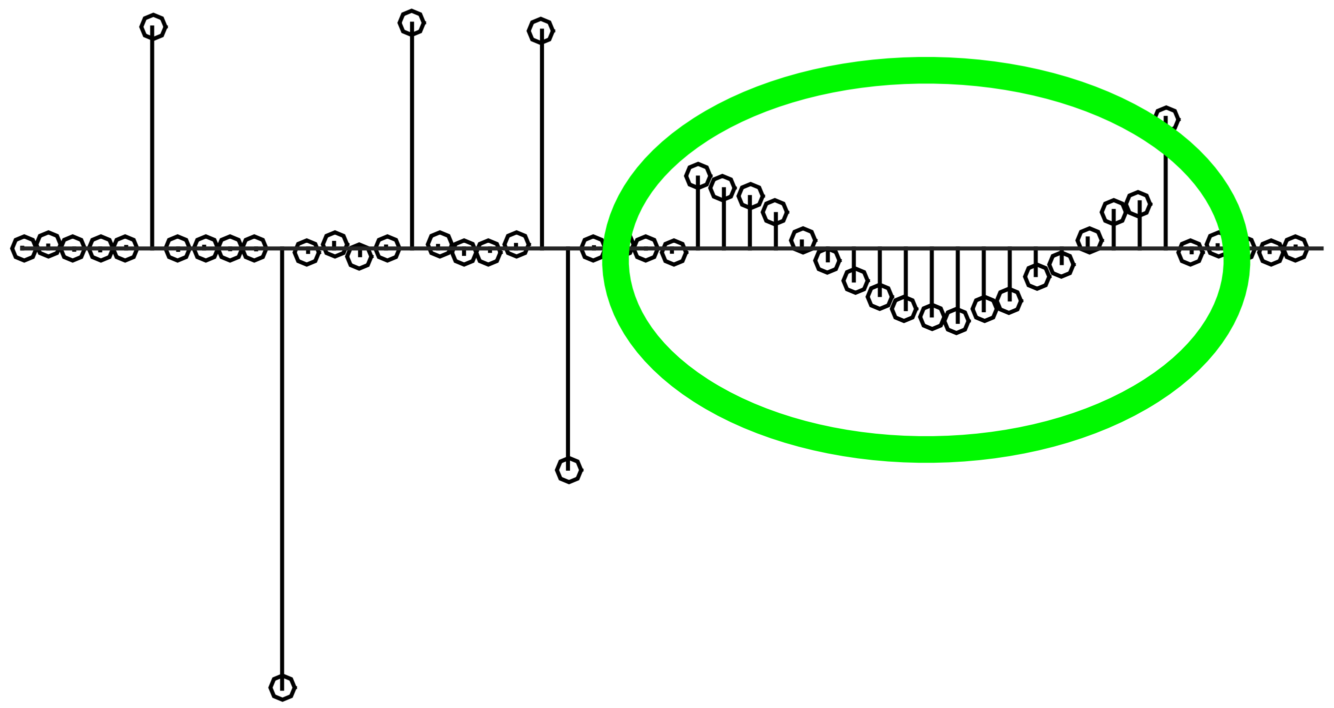


Example: Two-Point Moving Difference

$$x[t]$$
$$x[t-\Delta t]$$

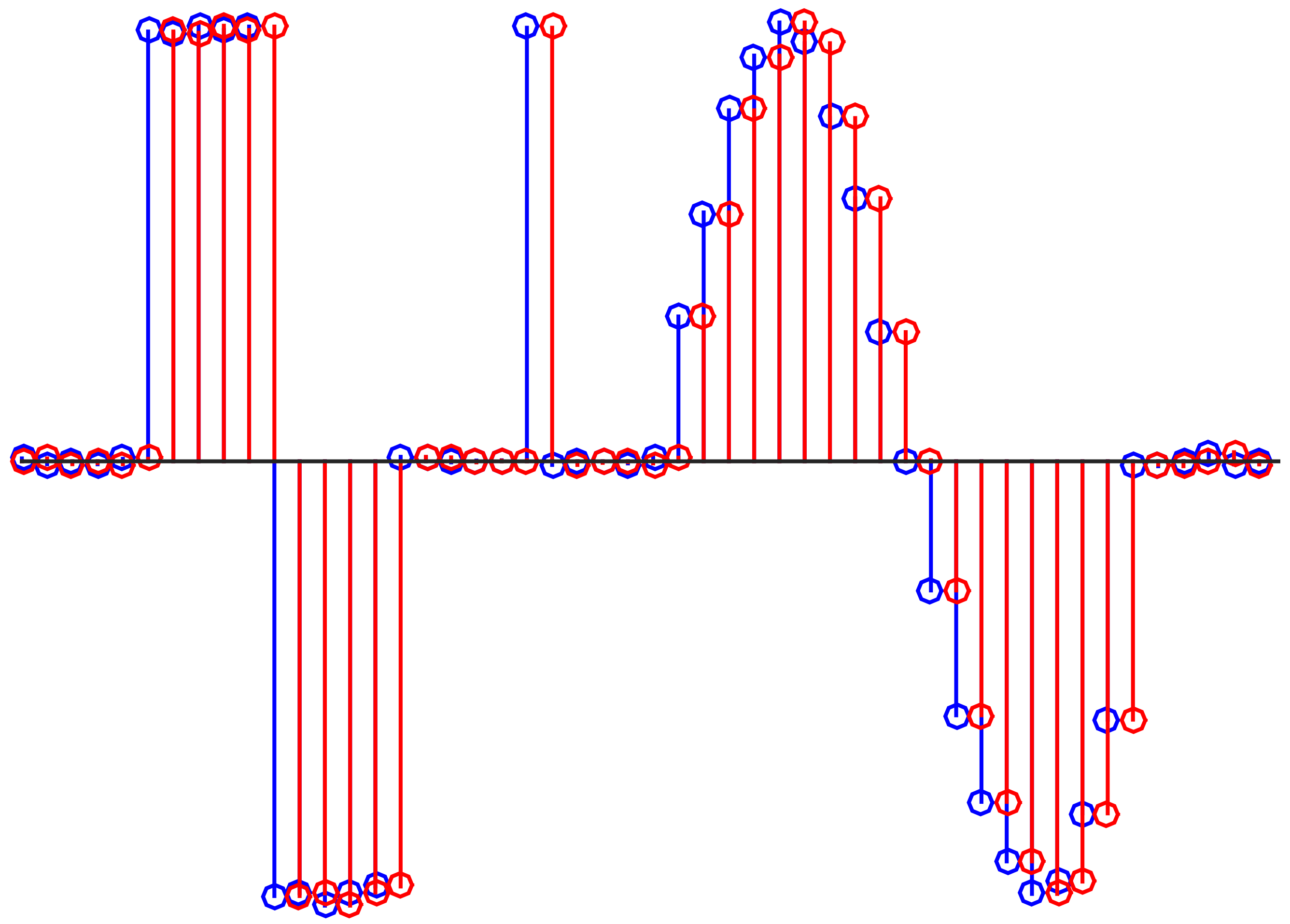


$$y[t]=\frac{1}{2}x[t]-\frac{1}{2}x[t-\Delta t]$$

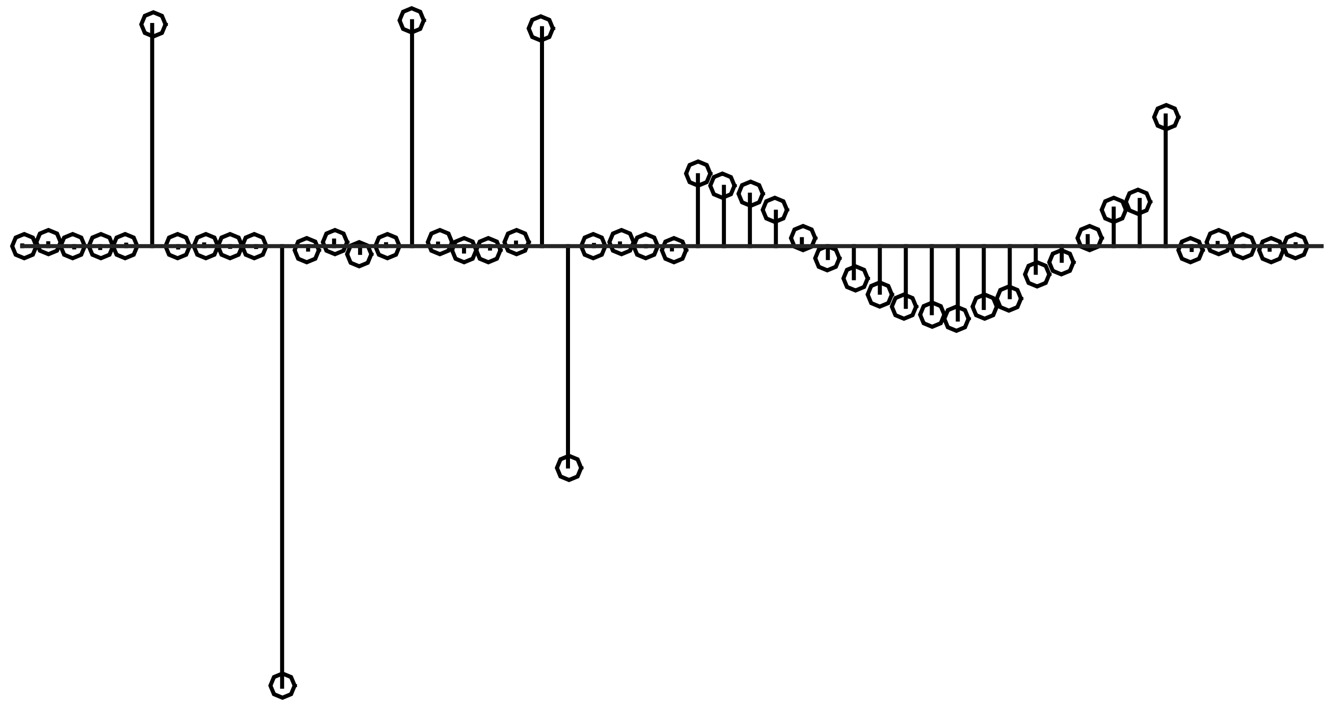


Example: Two-Point Moving Difference

$x[t]$
$x[t-\Delta t]$

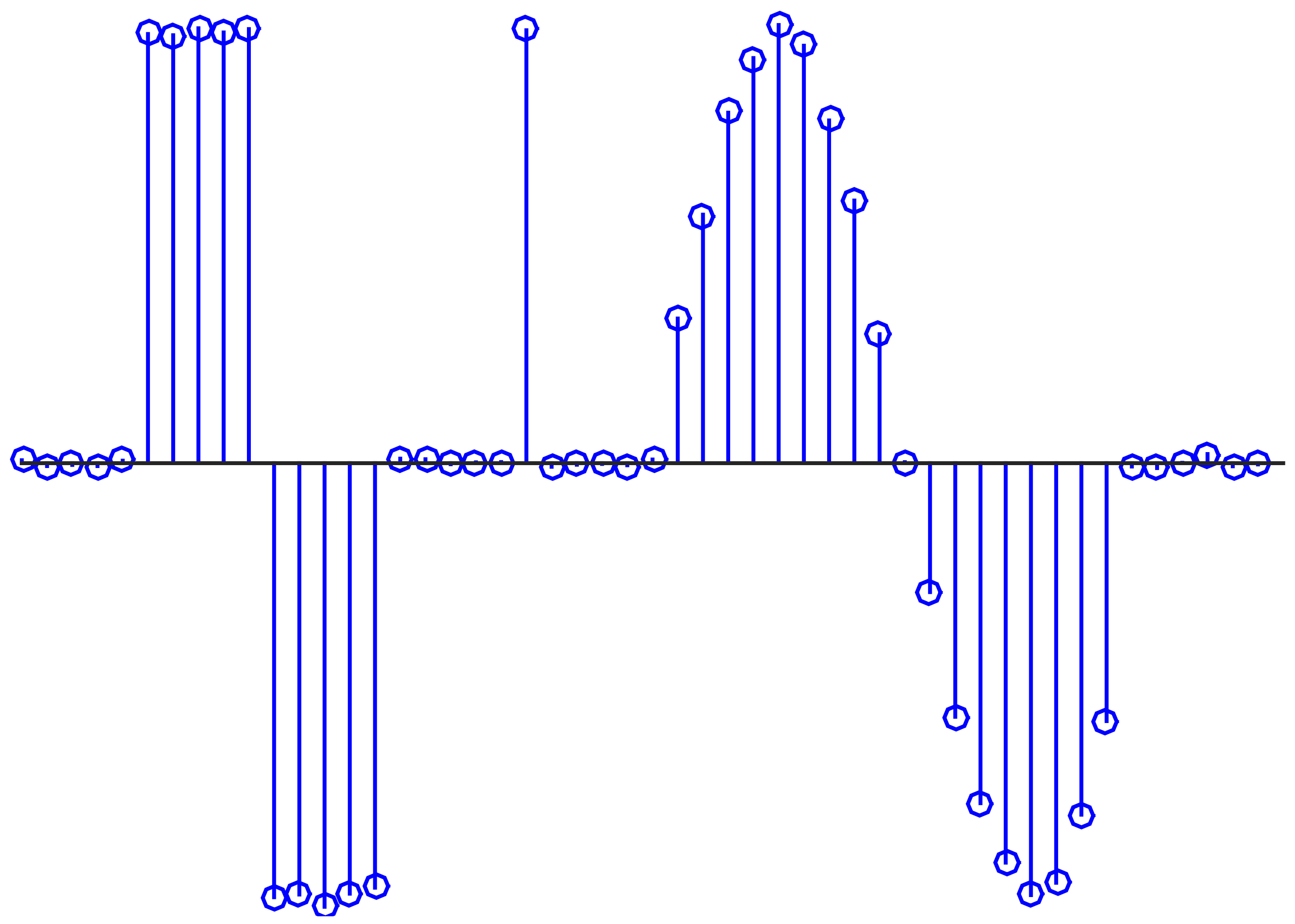


$$y[t] = \frac{1}{2}x[t] - \frac{1}{2}x[t-\Delta t]$$

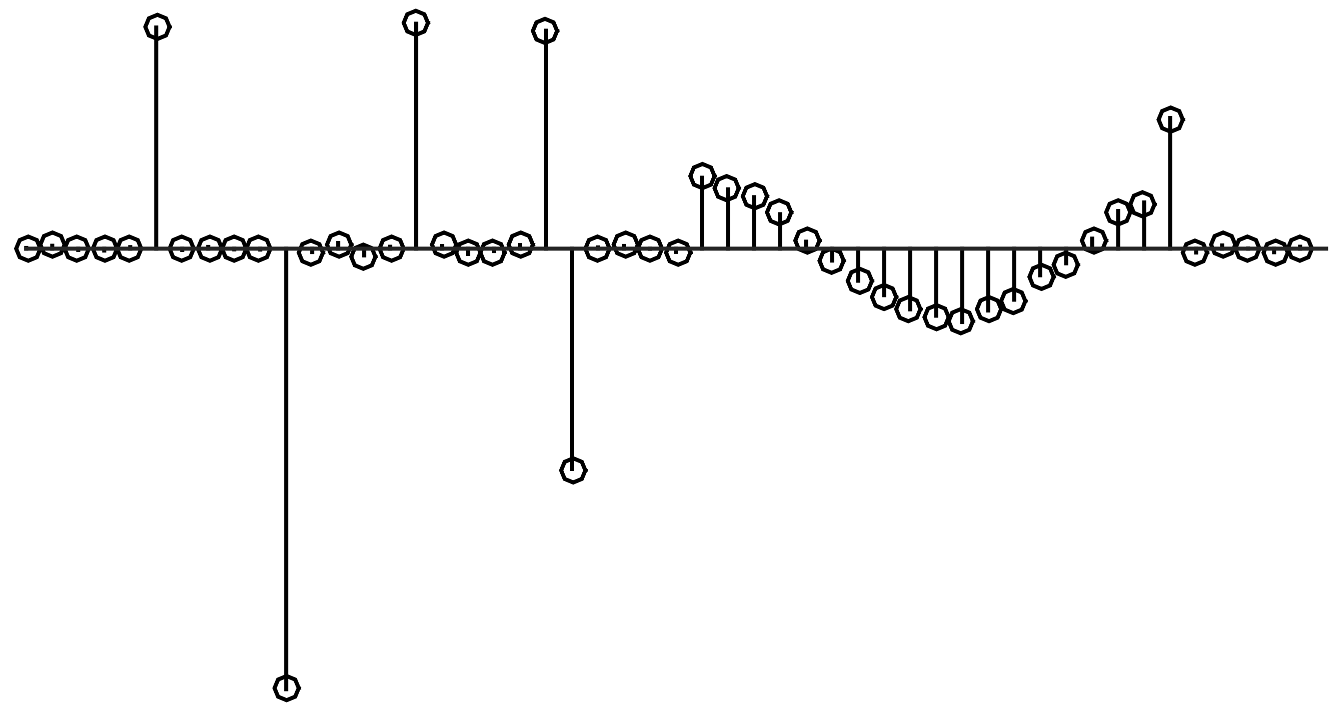


Example: Two-Point Moving Difference

$x[t]$



$$y[t] = \frac{1}{2}x[t] - \frac{1}{2}x[t - \Delta t]$$

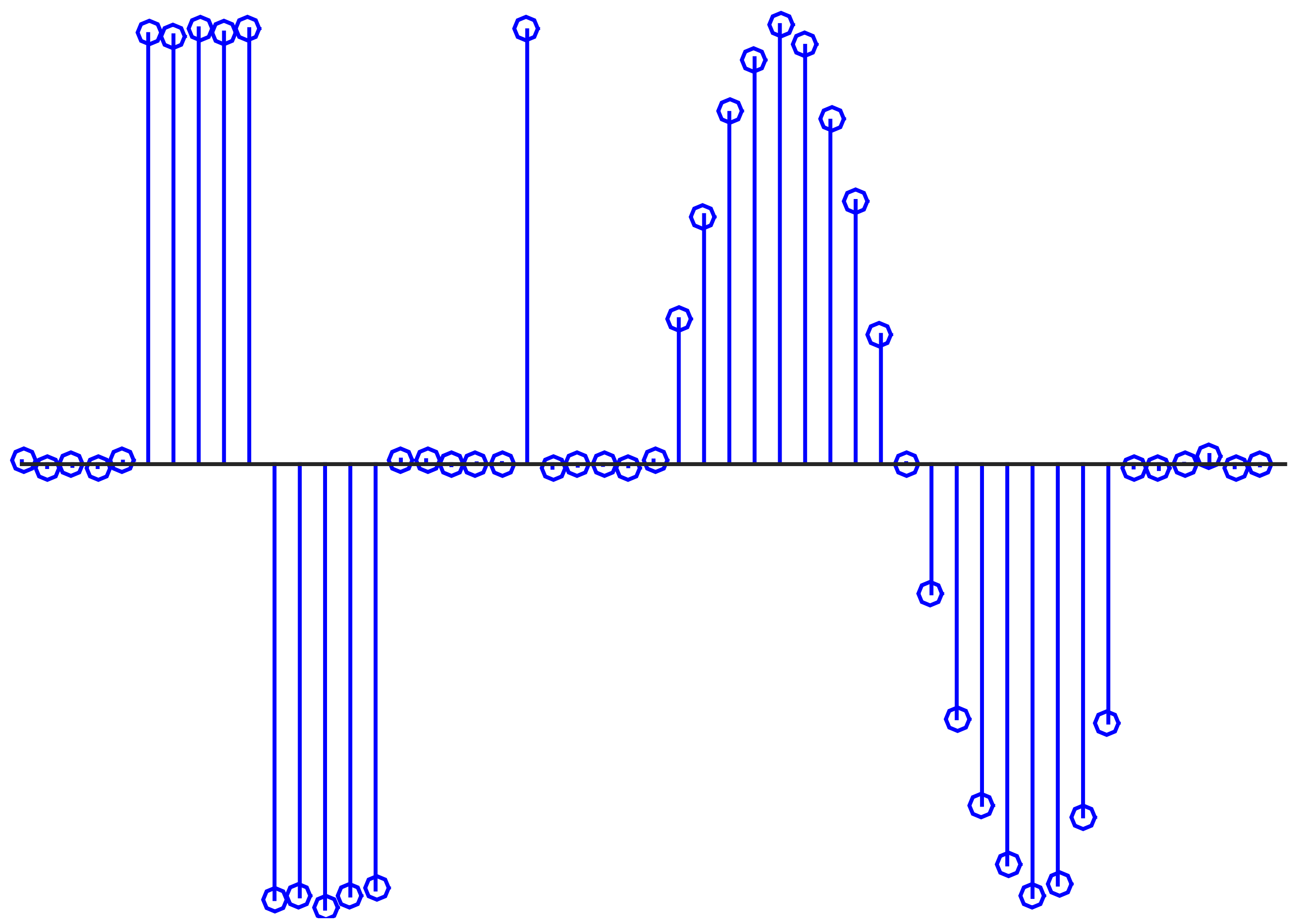


Results:

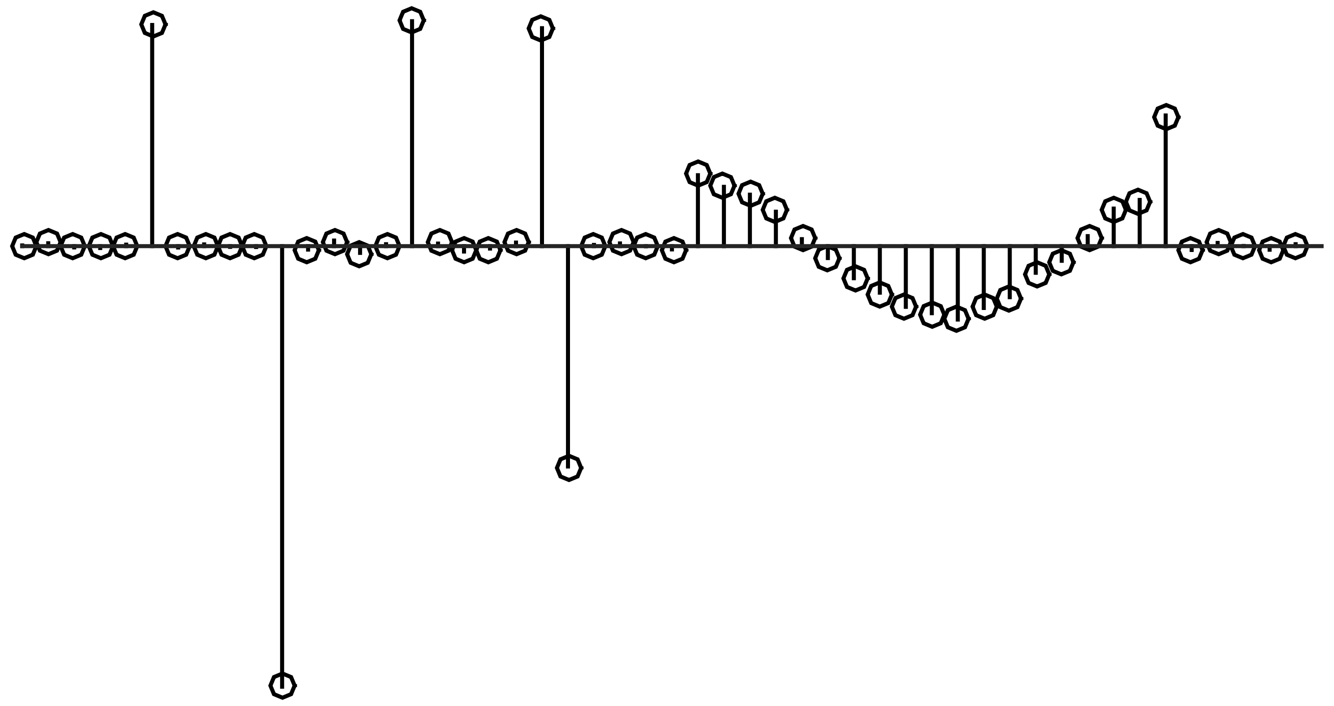
- Exaggerates differences
- Amplifies quickly varying signals
- Attenuates slowly varying signals
- High Pass Filter?

Example: Two-Point Moving Difference

$x[t]$



$$y[t] = \frac{1}{2}x[t] - \frac{1}{2}x[t - \Delta t]$$

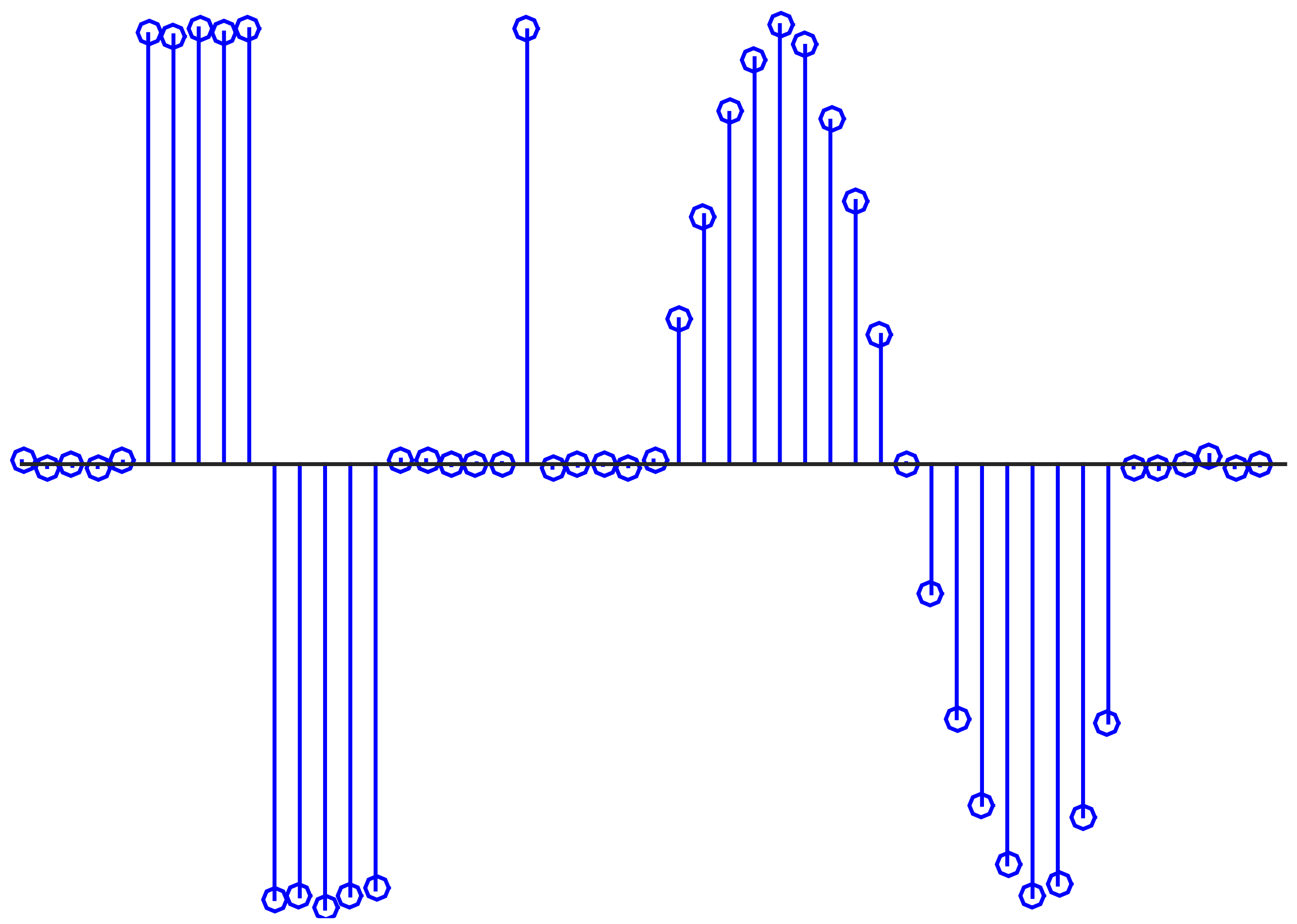


Results:

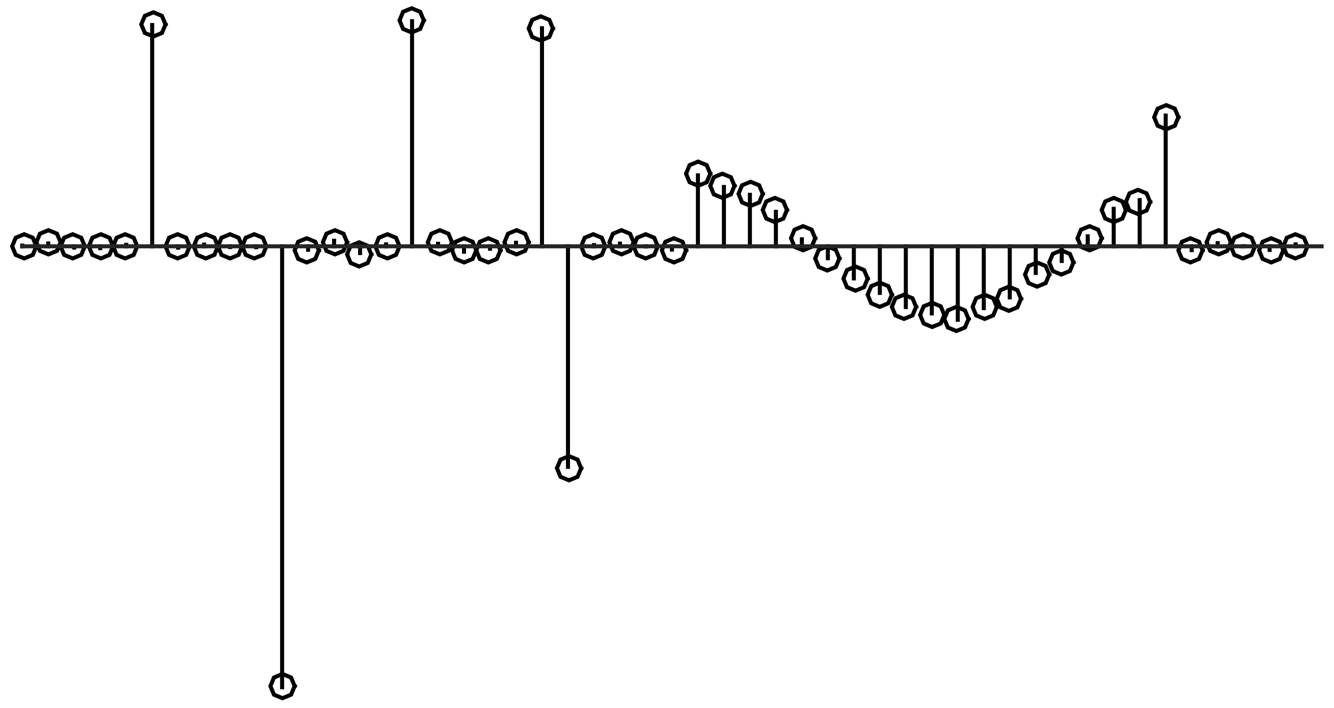
- Exaggerates differences
- Amplifies quickly varying signals
- Attenuates slowly varying signals
- High Pass Filter?

Break for Computer Lab Exercise 3

$x[t]$



$$y[t] = \frac{1}{2}x[t] - \frac{1}{2}x[t-\Delta t]$$



Results:

- Exaggerates differences
- Amplifies quickly varying signals
- Attenuates slowly varying signals
- High Pass Filter?

How do Filters affect Frequency?

- **Every** Time-Domain Signal can be Re-expressed as a Sum of Sinusoids/Oscillations
- # of time points = # of frequencies
- Reciprocal relationship: *time* resolution (Δt) & *frequency span* (f_s)
- Reciprocal relationship: *frequency resolution* (Δf) & *time span* (T)

$$x[t] = \frac{1}{N}\sum_{k=0}^{N-1} X[f_k]\mathrm{e}^{i2\pi f_k t}$$ where:

$$t = \underbrace{0, \Delta t, 2\Delta t, \ \ldots \ , T - \Delta t}_{N}$$

$$f_k = \underbrace{0, \Delta f, 2\Delta f, \ \ldots \ , f_s - \Delta f}_{N}$$

$$f_s = sampling\ frequency = \frac{1}{\Delta t}$$

$$T = signal\ duration = \frac{1}{\Delta f}$$

The Fourier Transform

- **Every** Time-Domain Signal can be Re-expressed as a Sum of Sinusoids/Oscillations
- # of time points = # of frequencies
- Reciprocal relationship: *time* resolution (Δt) & *frequency span* (f_s)
- Reciprocal relationship: *frequency resolution* (Δf) & *time span* (T)

$$x[t] = \frac{1}{N}\sum_{k=0}^{N-1} X[f_k]\mathrm{e}^{i2\pi f_k t} \quad \text{where:}$$

$$t = \underbrace{0, \Delta t, 2\Delta t, \ \ldots \ , T - \Delta t}_{N}$$

$$f_k = \underbrace{0, \Delta f, 2\Delta f, \ \ldots \ , f_s - \Delta f}_{N}$$

$$f_s = sampling\ frequency = \frac{1}{\Delta t}$$

$$T = signal\ duration = \frac{1}{\Delta f}$$

The Fourier Transform

- **Every** Time-Domain Signal can be Re-expressed as a Sum of Sinusoids/Oscillations
- # of time points = # of frequencies
- Reciprocal relationship: *time* resolution (Δt) & *frequency span* (f_s)
- Reciprocal relationship: *frequency resolution* (Δf) & *time span* (T)

$$x[t] = \frac{1}{N}\sum_{k=0}^{N-1} X[f_k]\mathrm{e}^{i2\pi f_k t}$$ where:

$$t = \underbrace{0, \Delta t, 2\Delta t, \ \ldots \ , T - \Delta t}_{N}$$

$$f_k = \underbrace{0, \Delta f, 2\Delta f, \ \ldots \ , f_s - \Delta f}_{N}$$

$$f_s = sampling\ frequency = \frac{1}{\Delta t}$$

$$T = signal\ duration = \frac{1}{\Delta f}$$

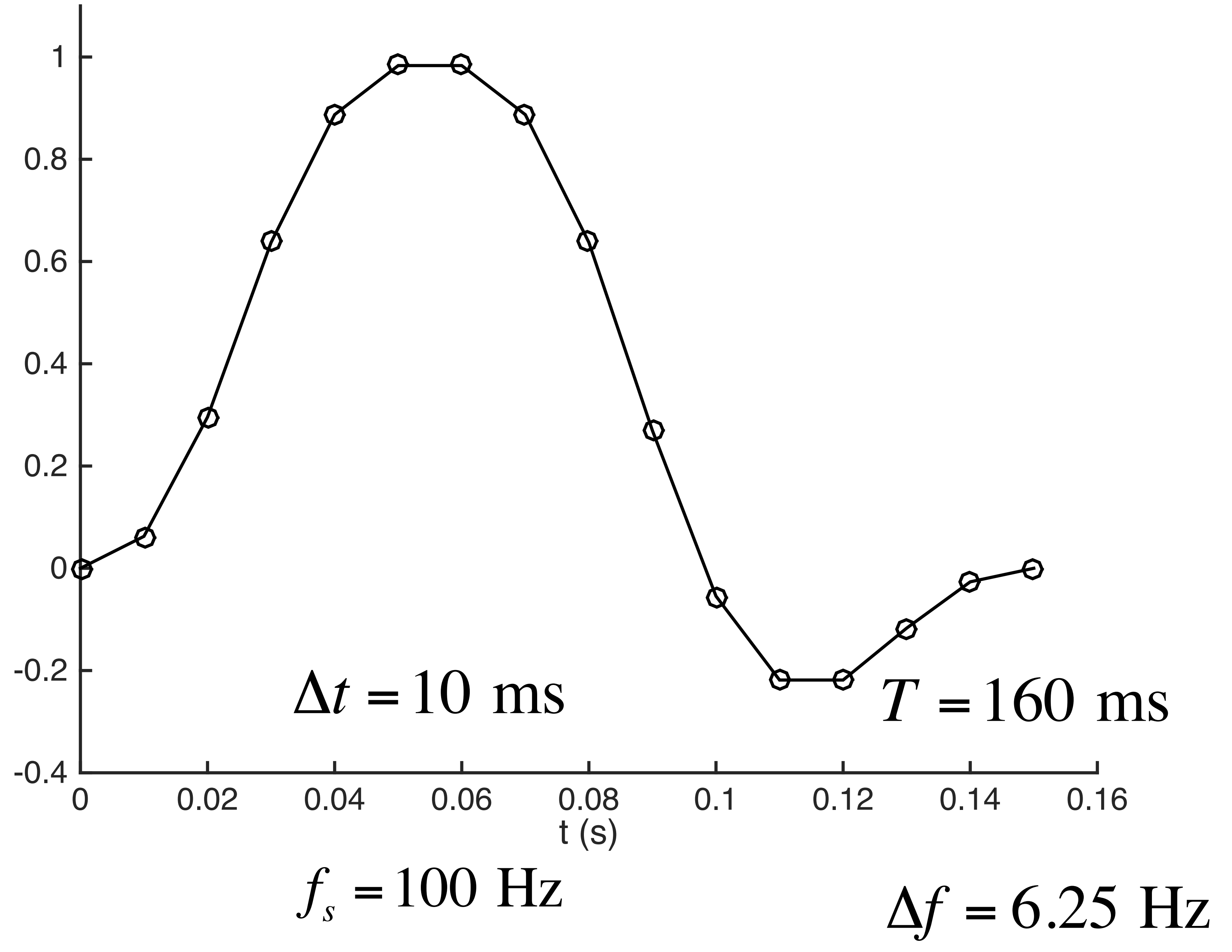
The Fourier Transform

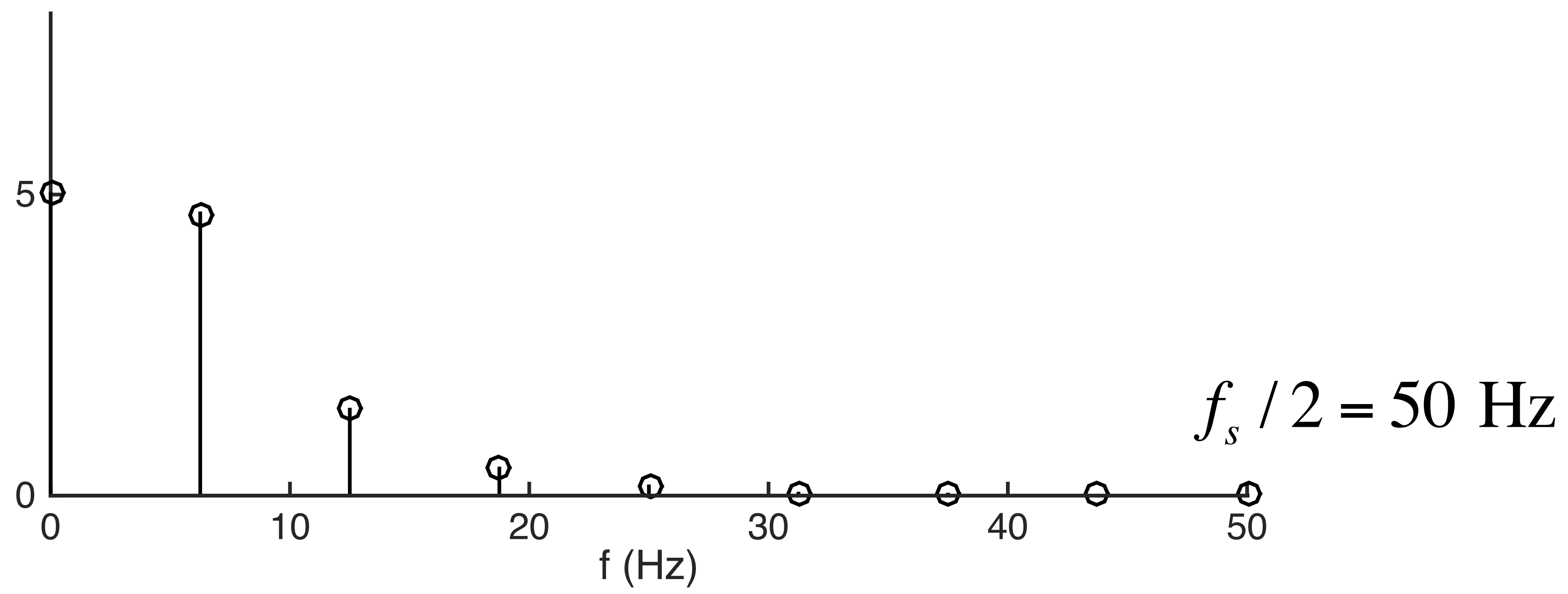
- **Every** Time-Domain Signal can be Re-expressed as a Sum of Sinusoids/Oscillations

- # of time points = # of frequencies

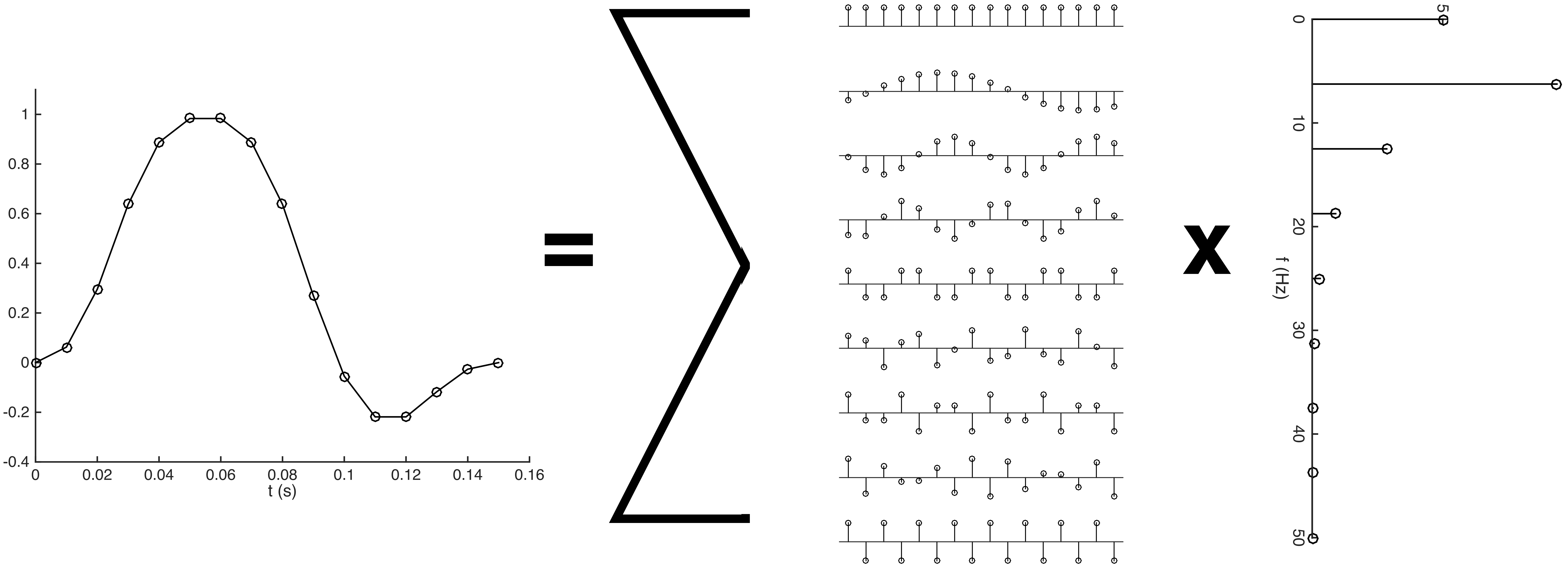
- Reciprocal relationship: *time resolution* (Δt) & *frequency span* (f_s)

- Reciprocal relationship: *frequency resolution* (Δf) & *time span* (T)

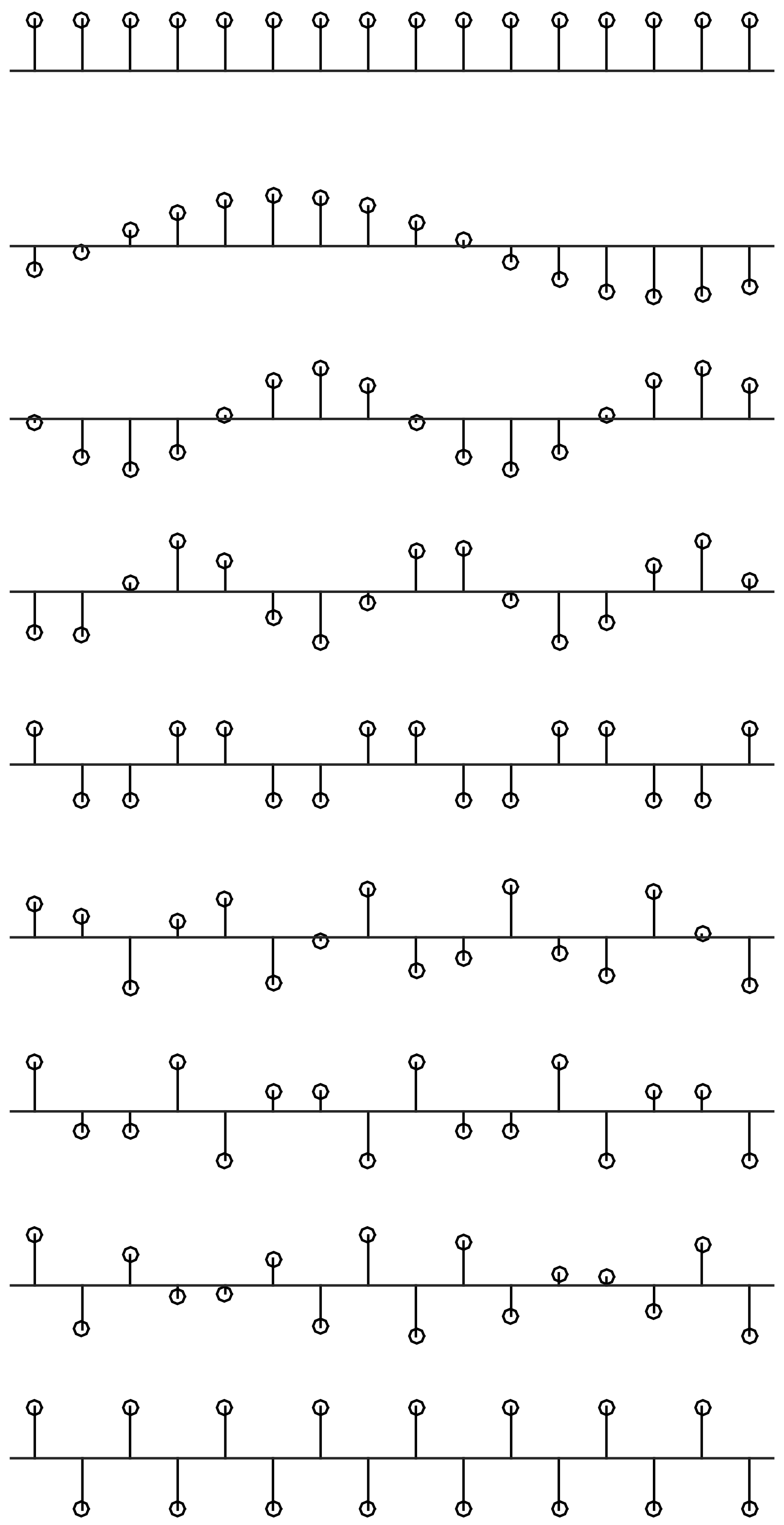




The Fourier Transform

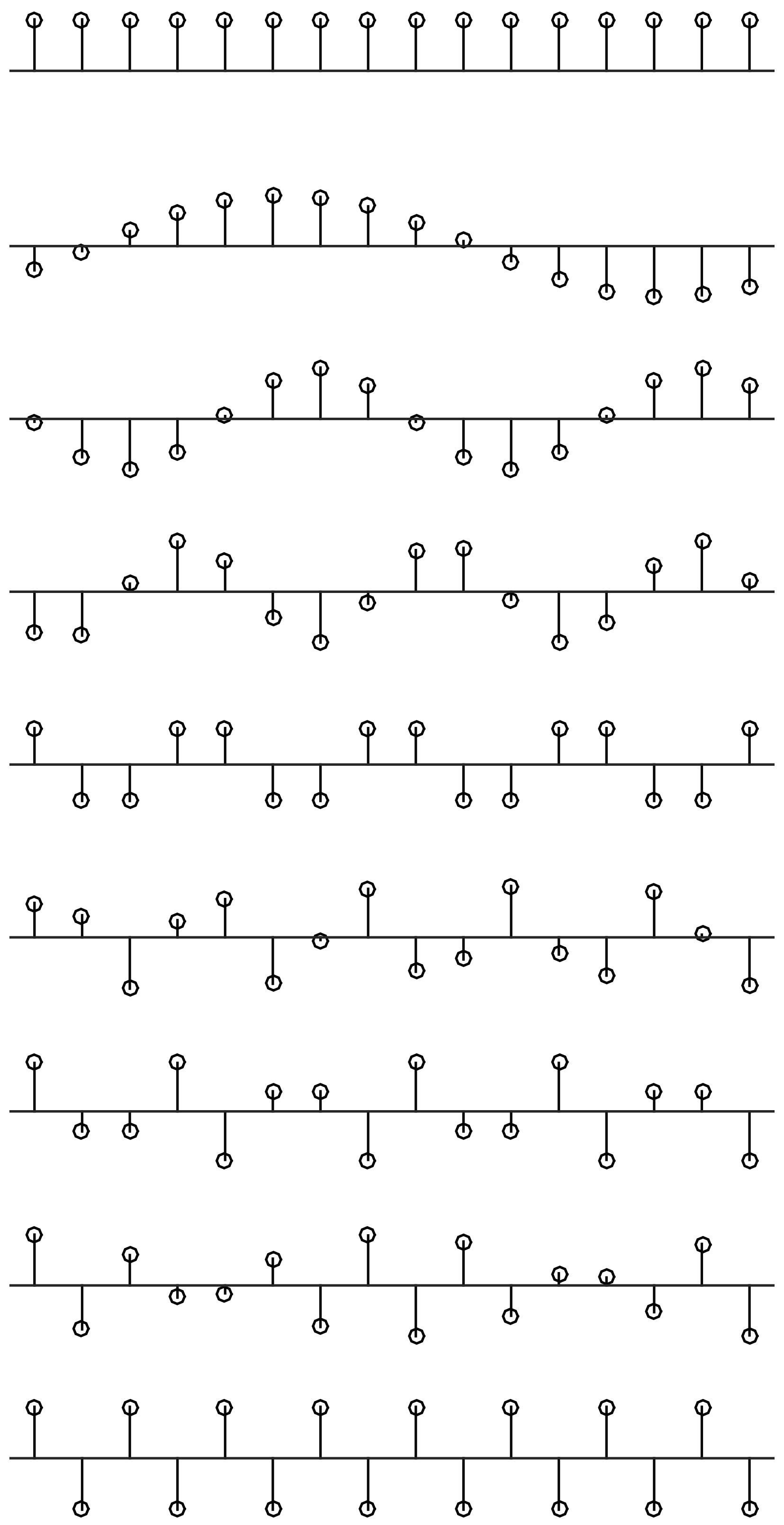


The Fourier Transform

Every Signal = $\sum$  x Fourier Coefficients

Filters and the Fourier Transform

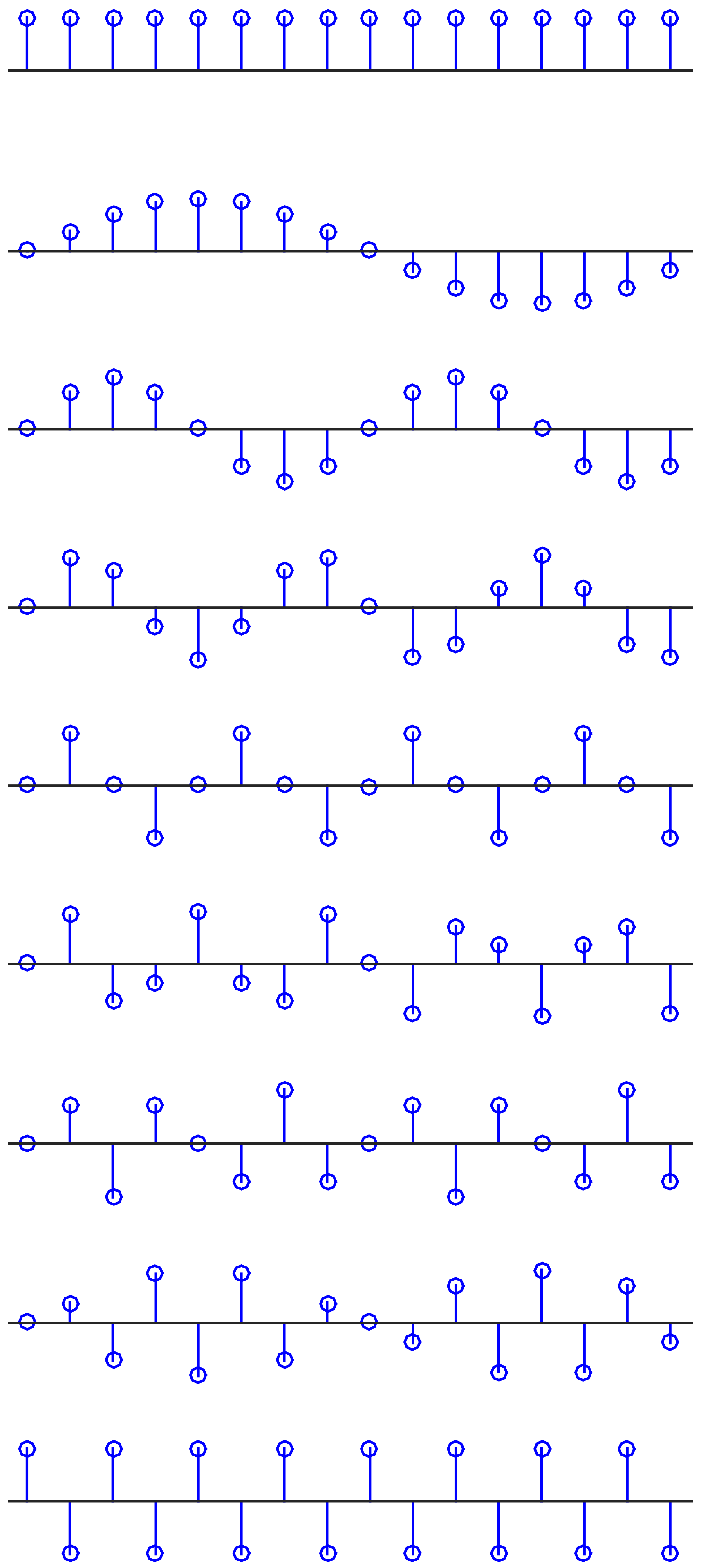
Filter(signal) **=** $\sum$ Filter(



) **x** Fourier Coefficients

Filters and the Fourier Transform

So it's really important what the filter does to these:

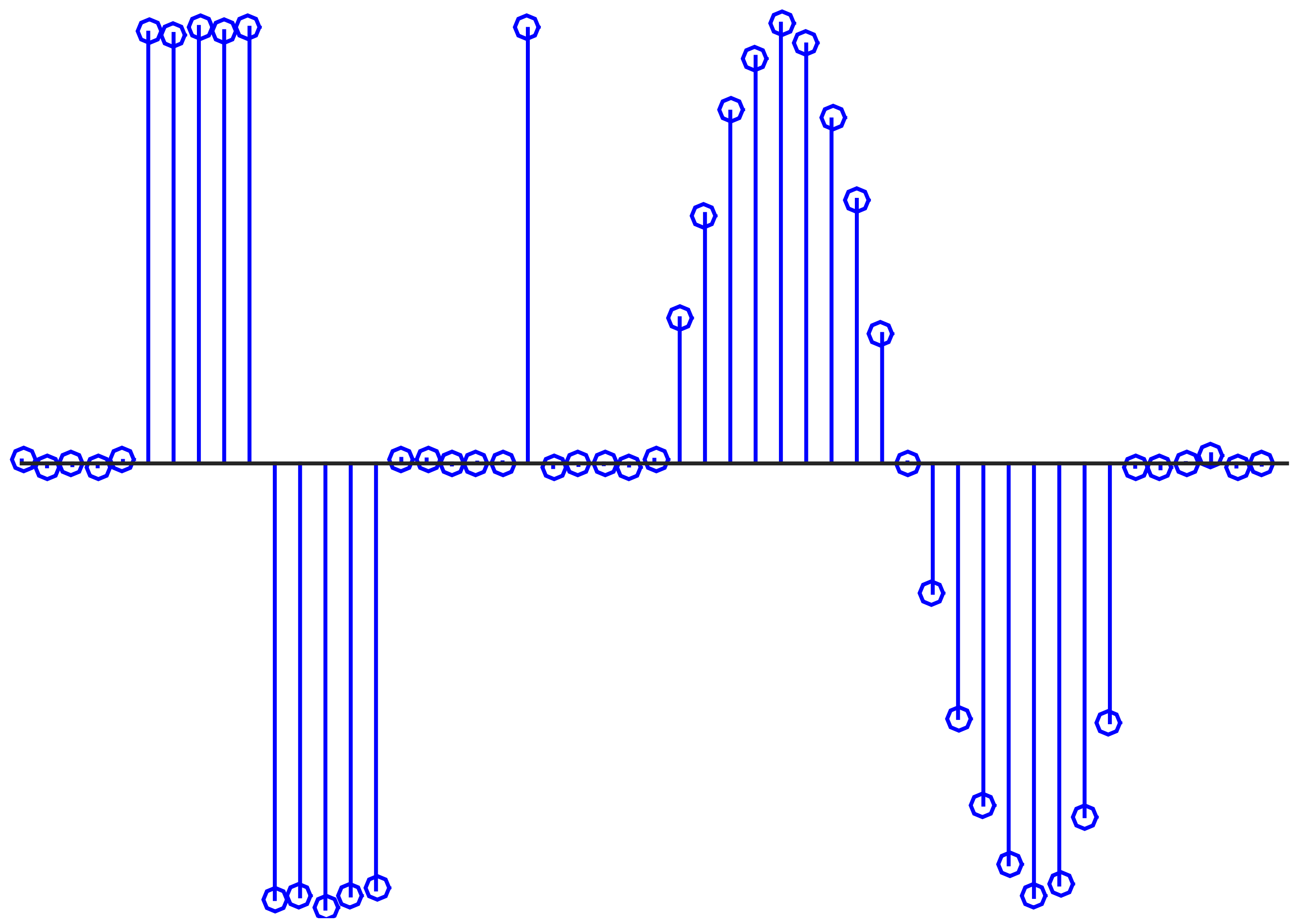


$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t - \Delta t]$$

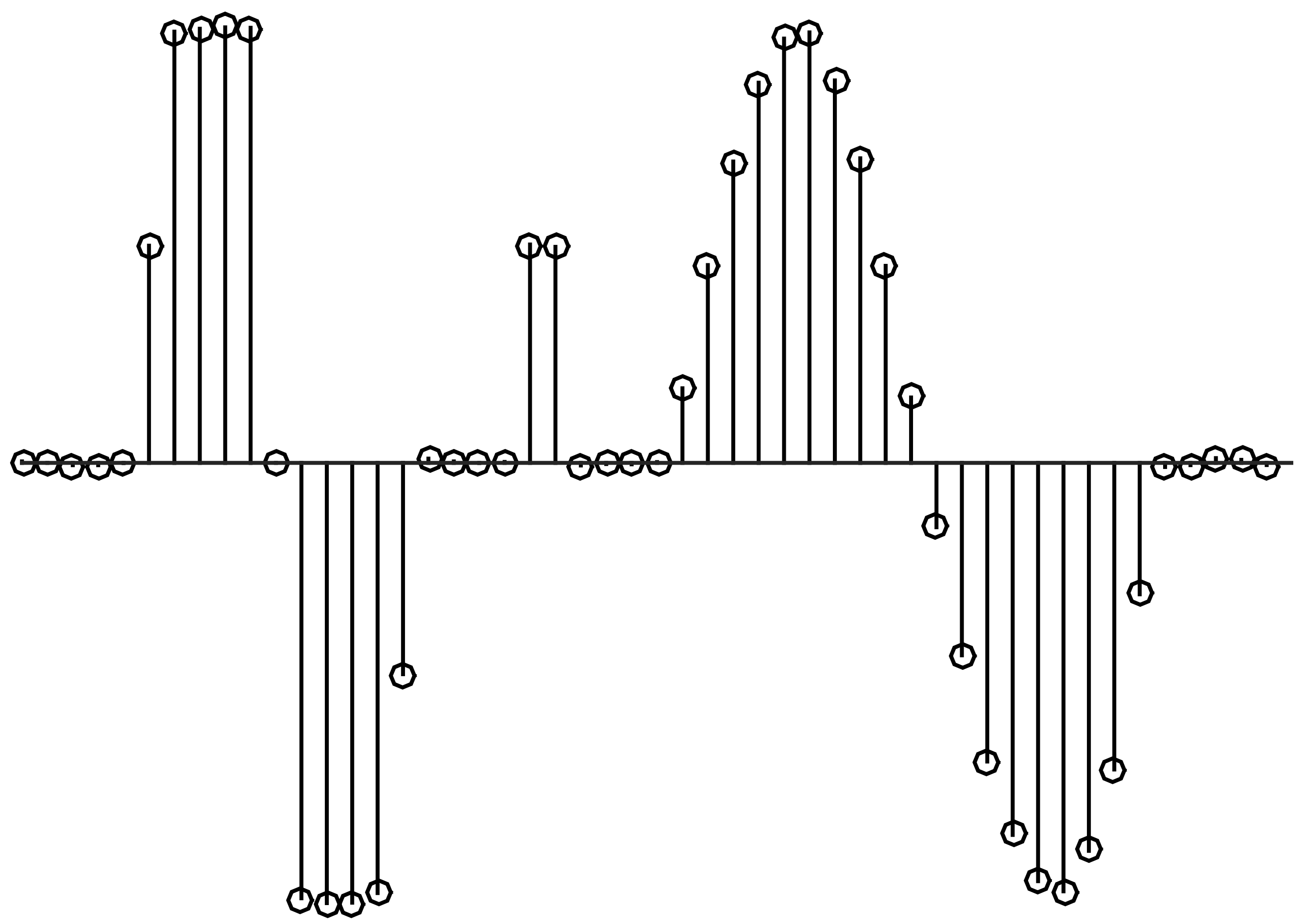
?

Filters and the Fourier Transform

Recall:

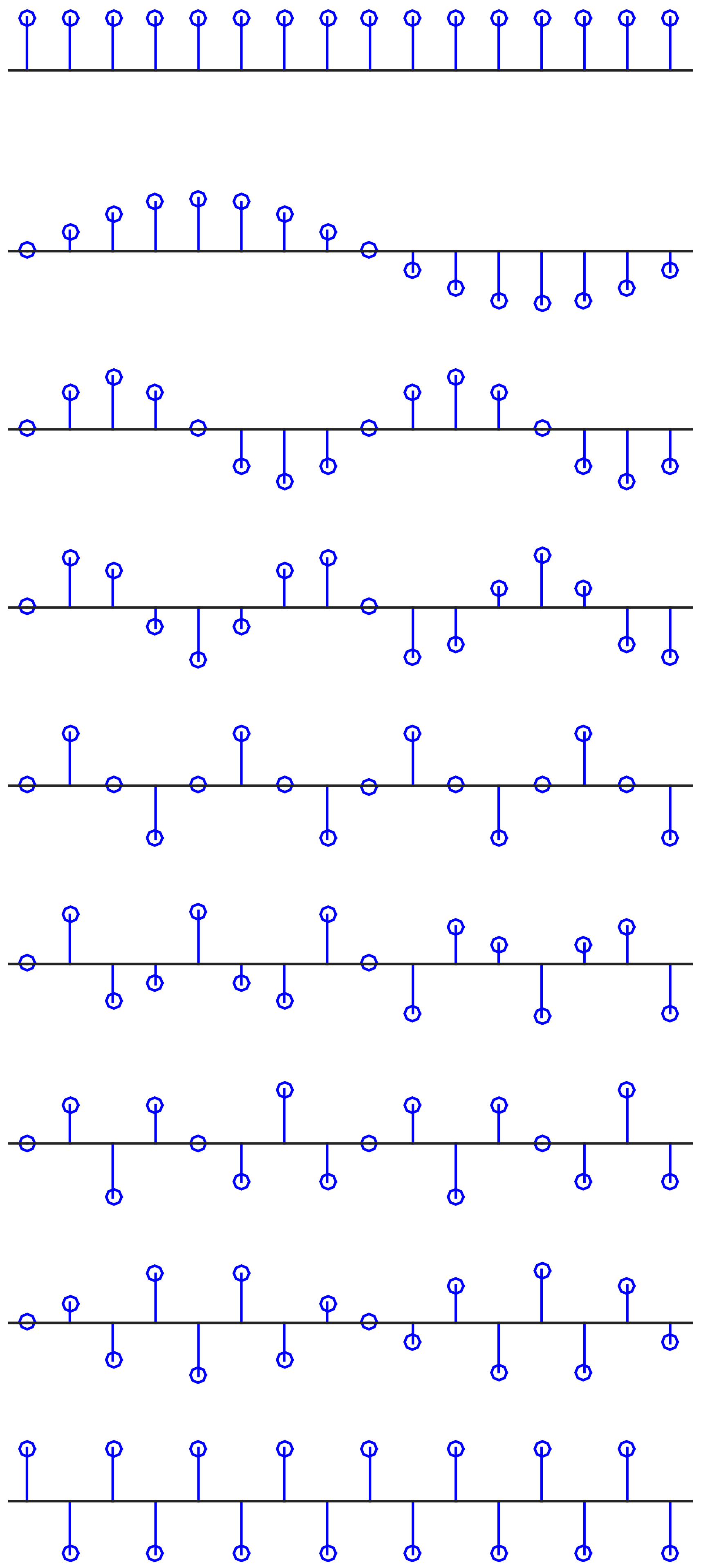


$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$



Filters and the Fourier Transform

So it's really important what the filter does to these:

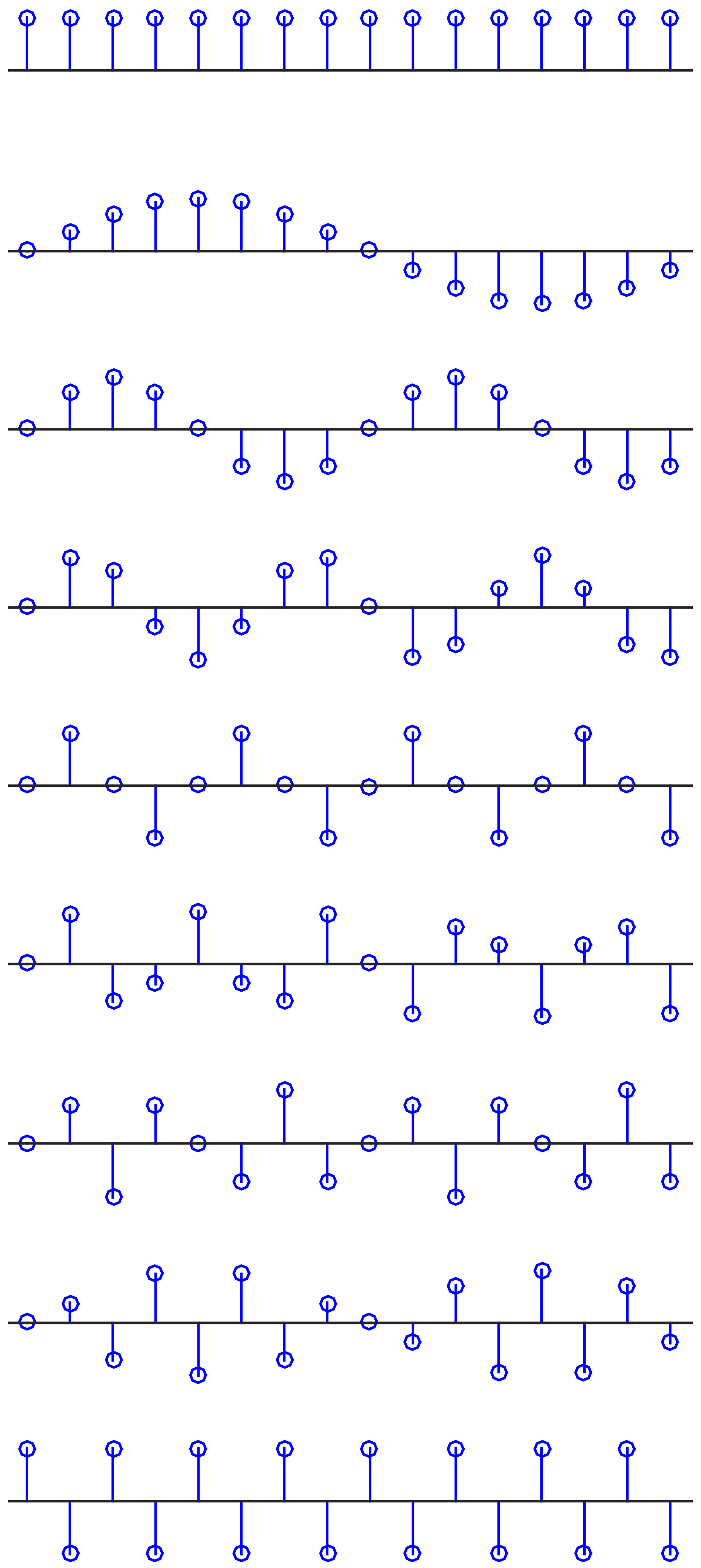


$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$

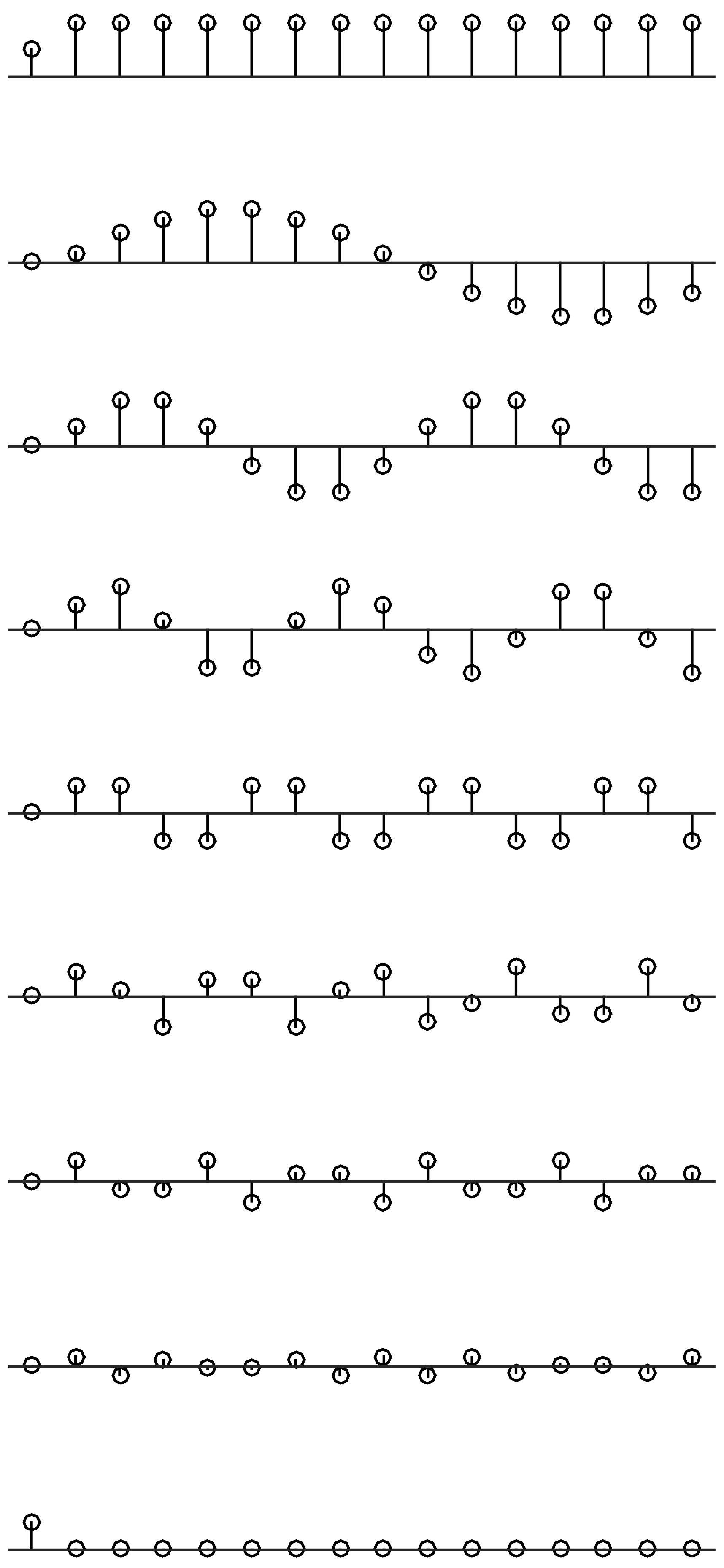
?

Filters and the Fourier Transform

So it's really important what the filter does to these:

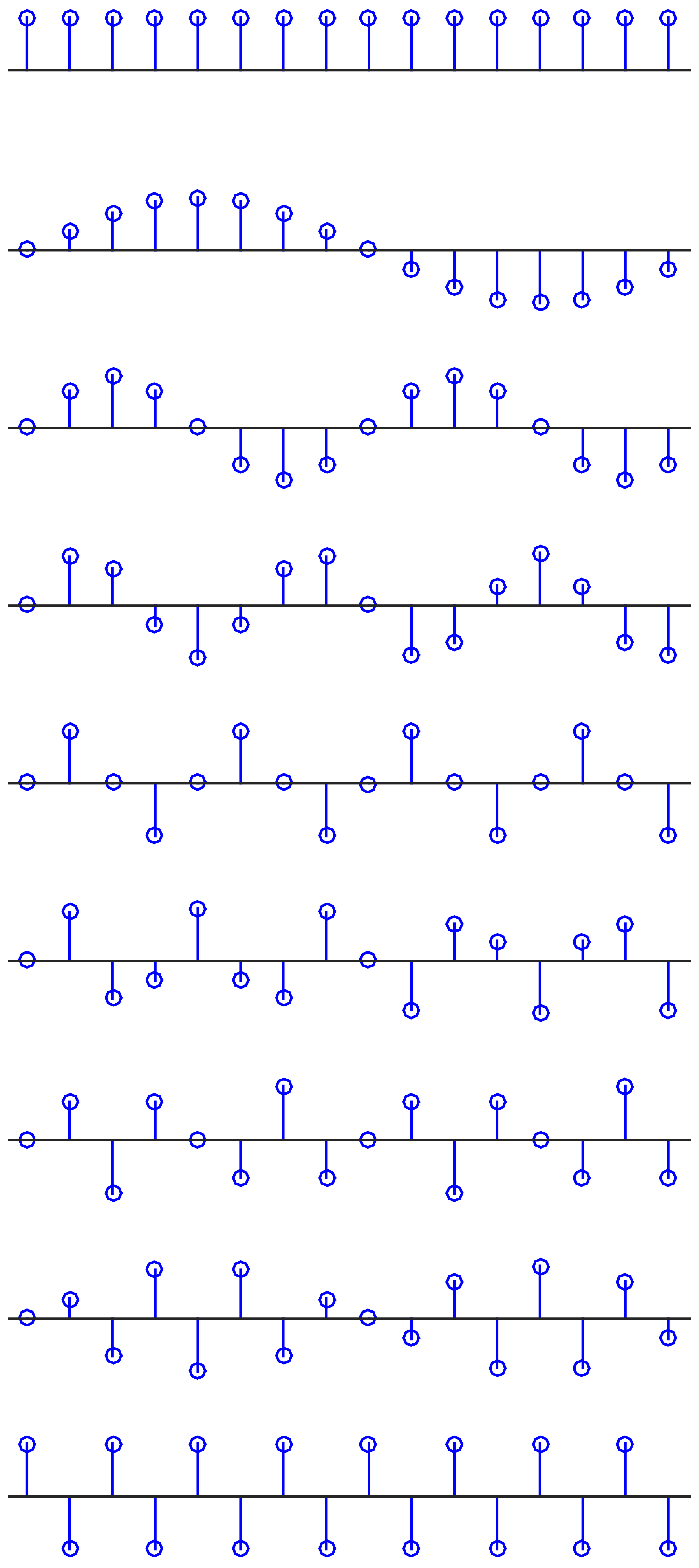


$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$

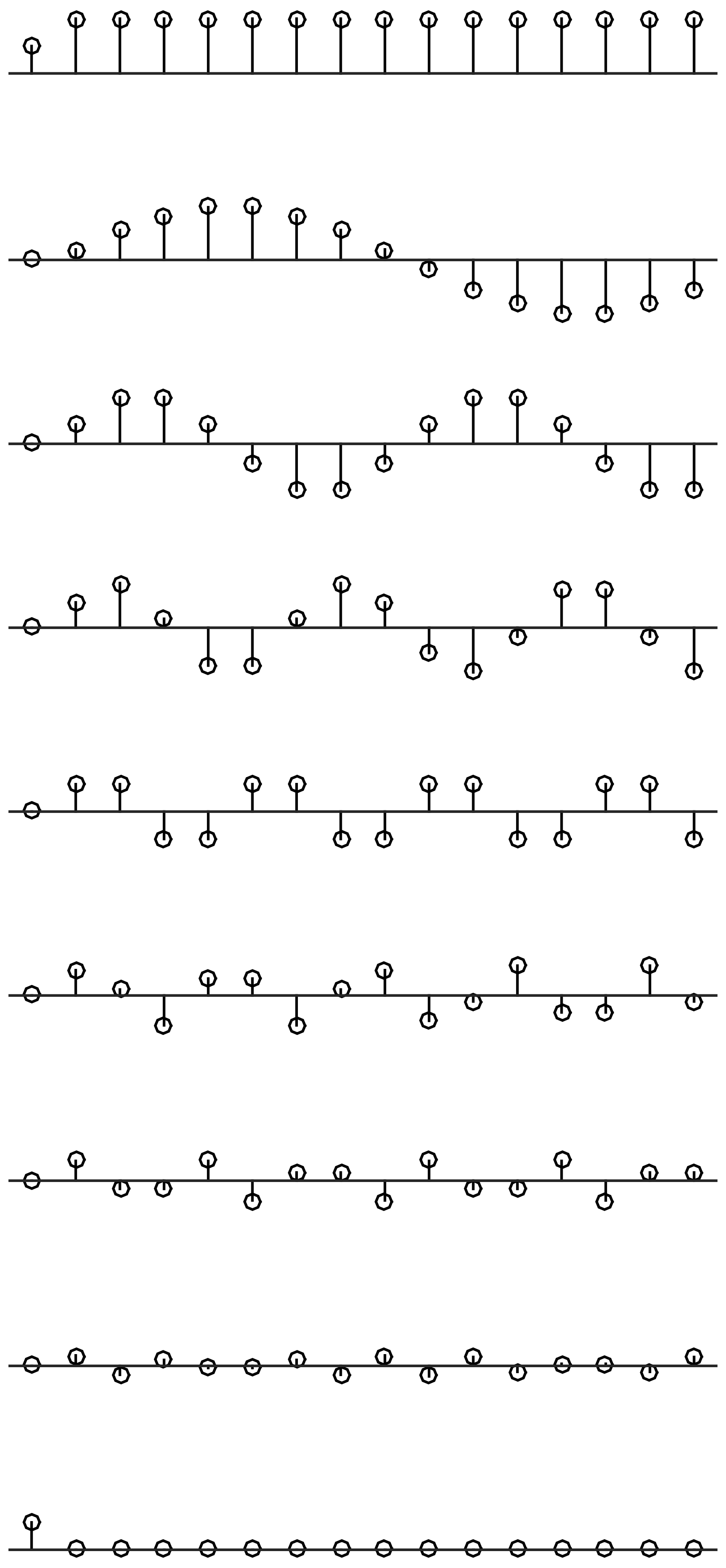


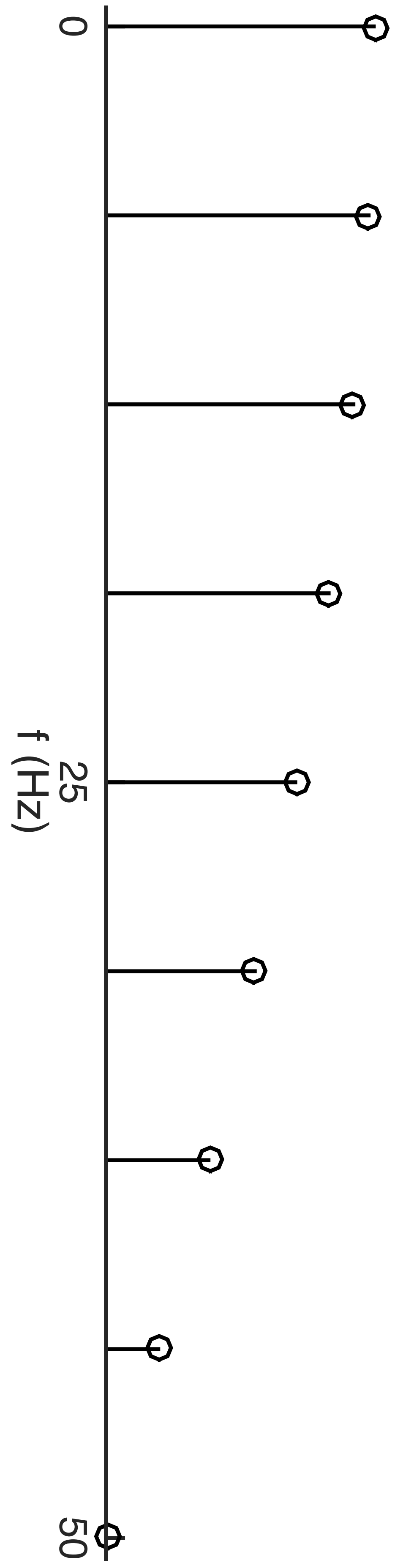
Filters and the Fourier Transform

So it's really important what the filter does to these:



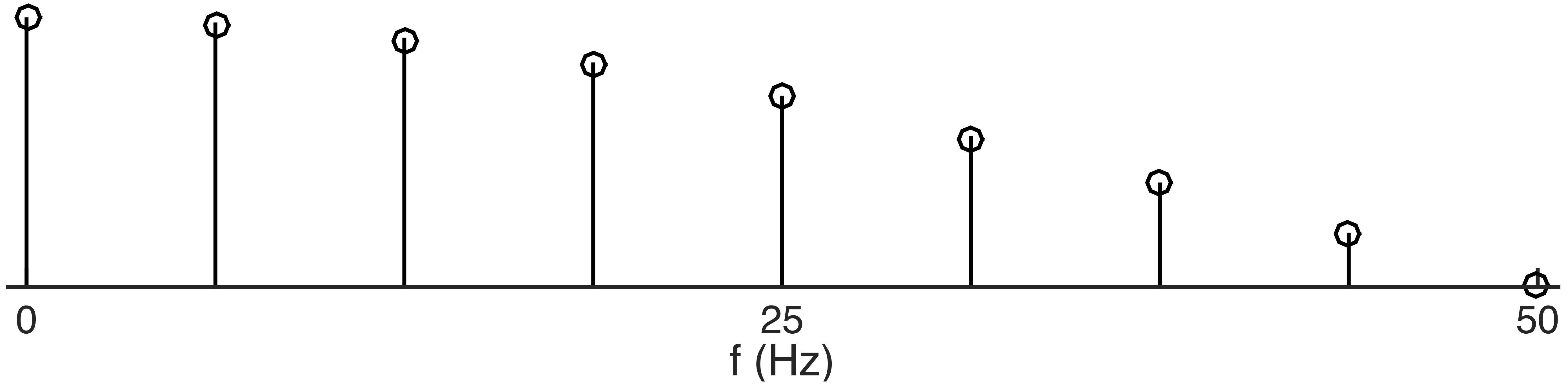
$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t - \Delta t]$$





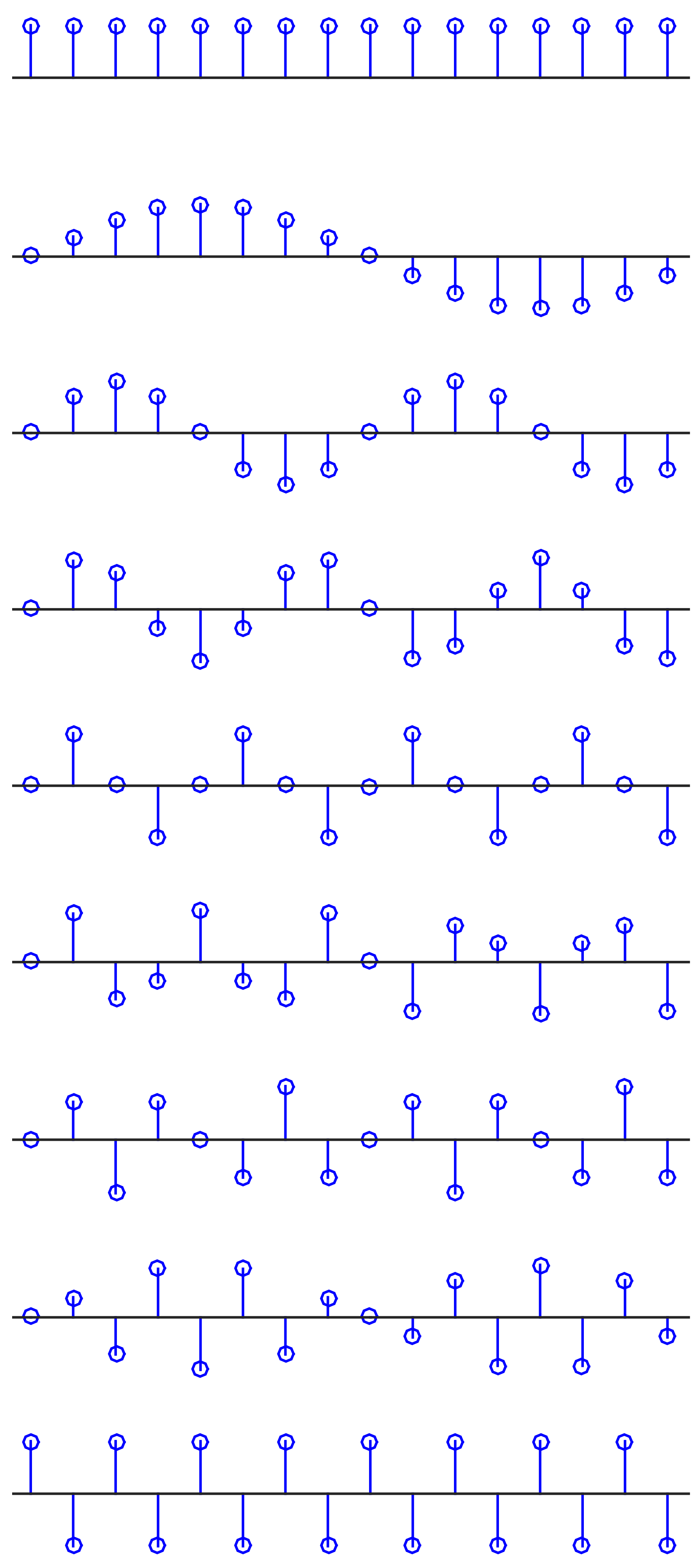
Filters and the Fourier Transform

$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t - \Delta t]$$



Low Pass Filter

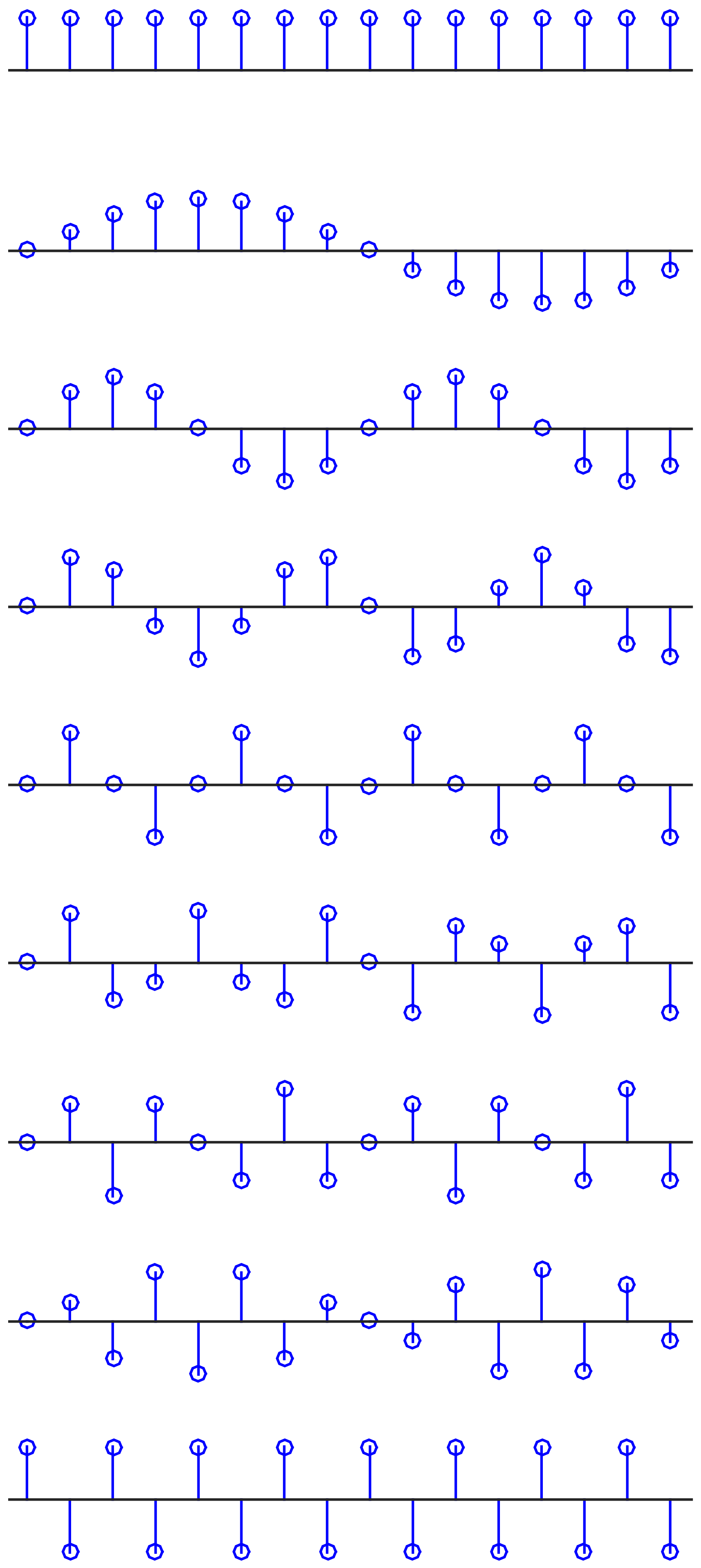
Filters and the Fourier Transform



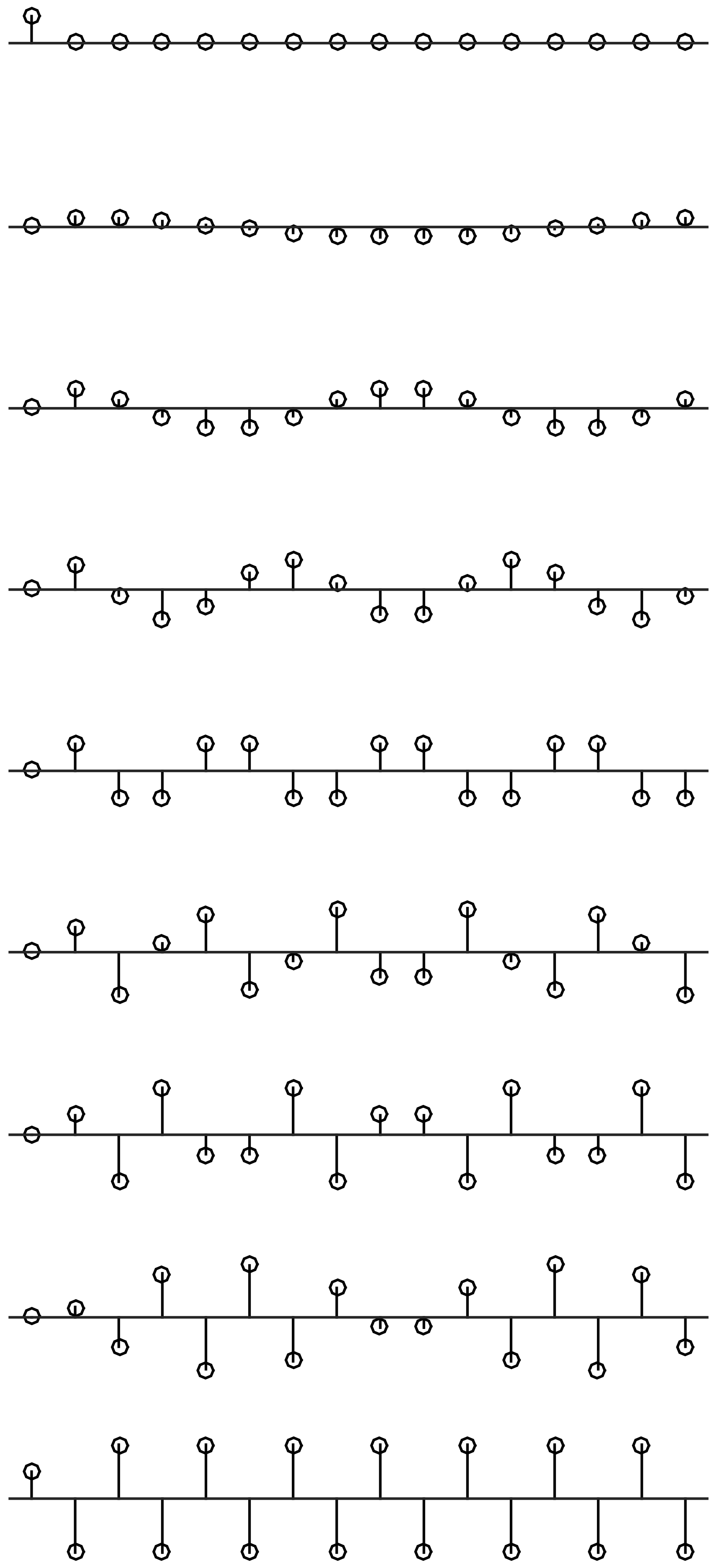
$$y[t] = \frac{1}{2}x[t] - \frac{1}{2}x[t - \Delta t]$$

→ **?**

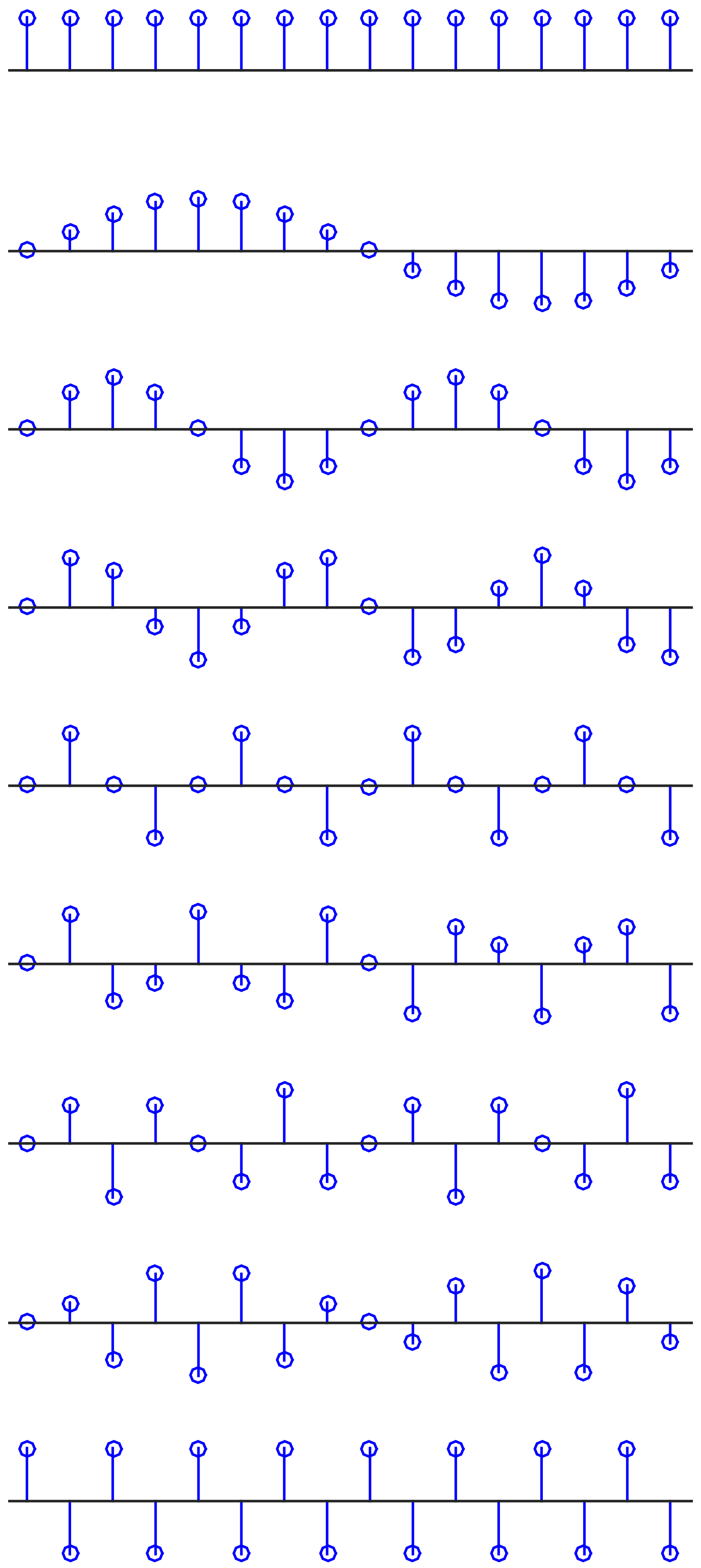
Filters and the Fourier Transform



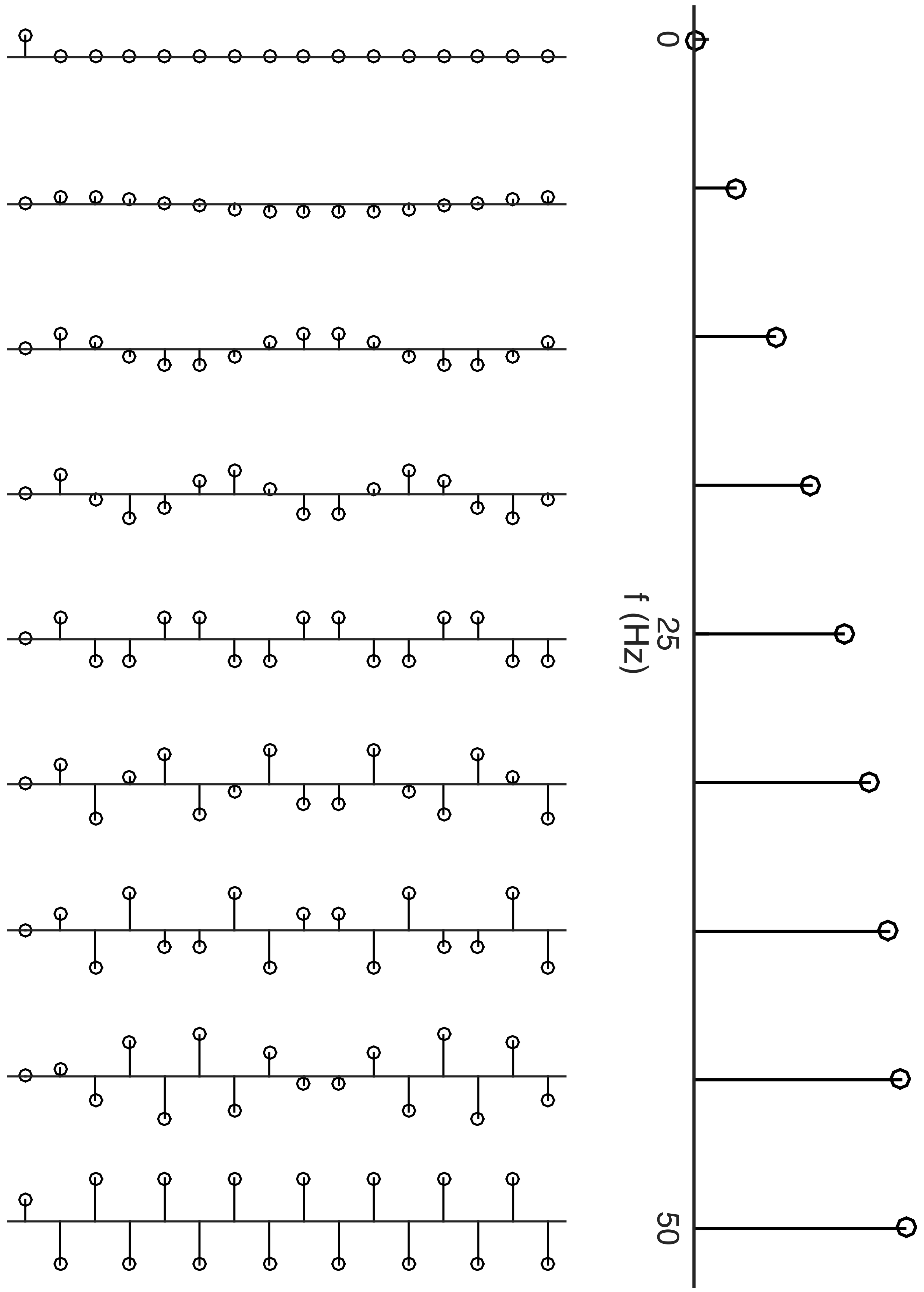
$$y[t] = \frac{1}{2}x[t] - \frac{1}{2}x[t - \Delta t]$$



Filters and the Fourier Transform

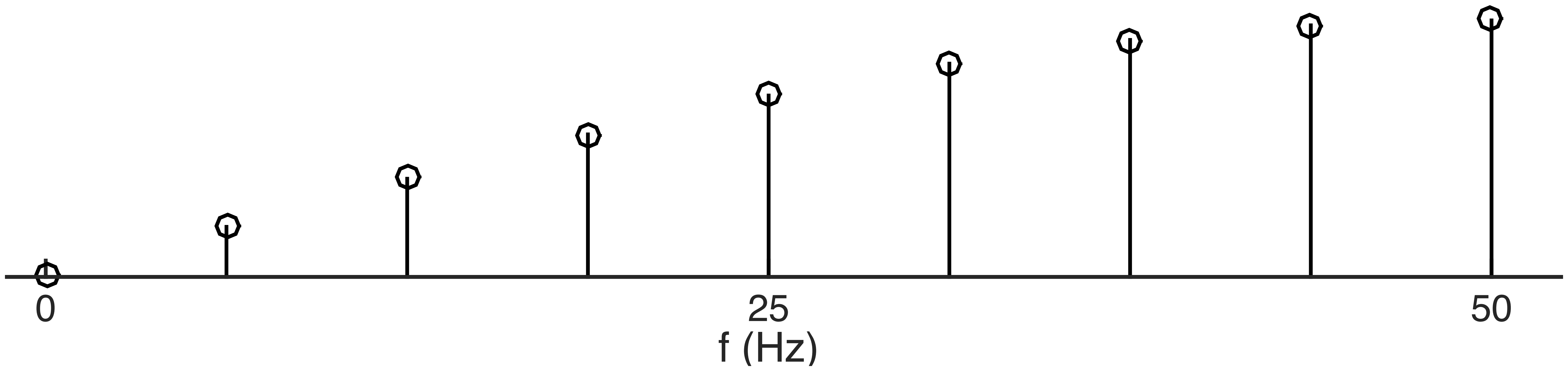


$$y[t] = \frac{1}{2}x[t] - \frac{1}{2}x[t - \Delta t]$$



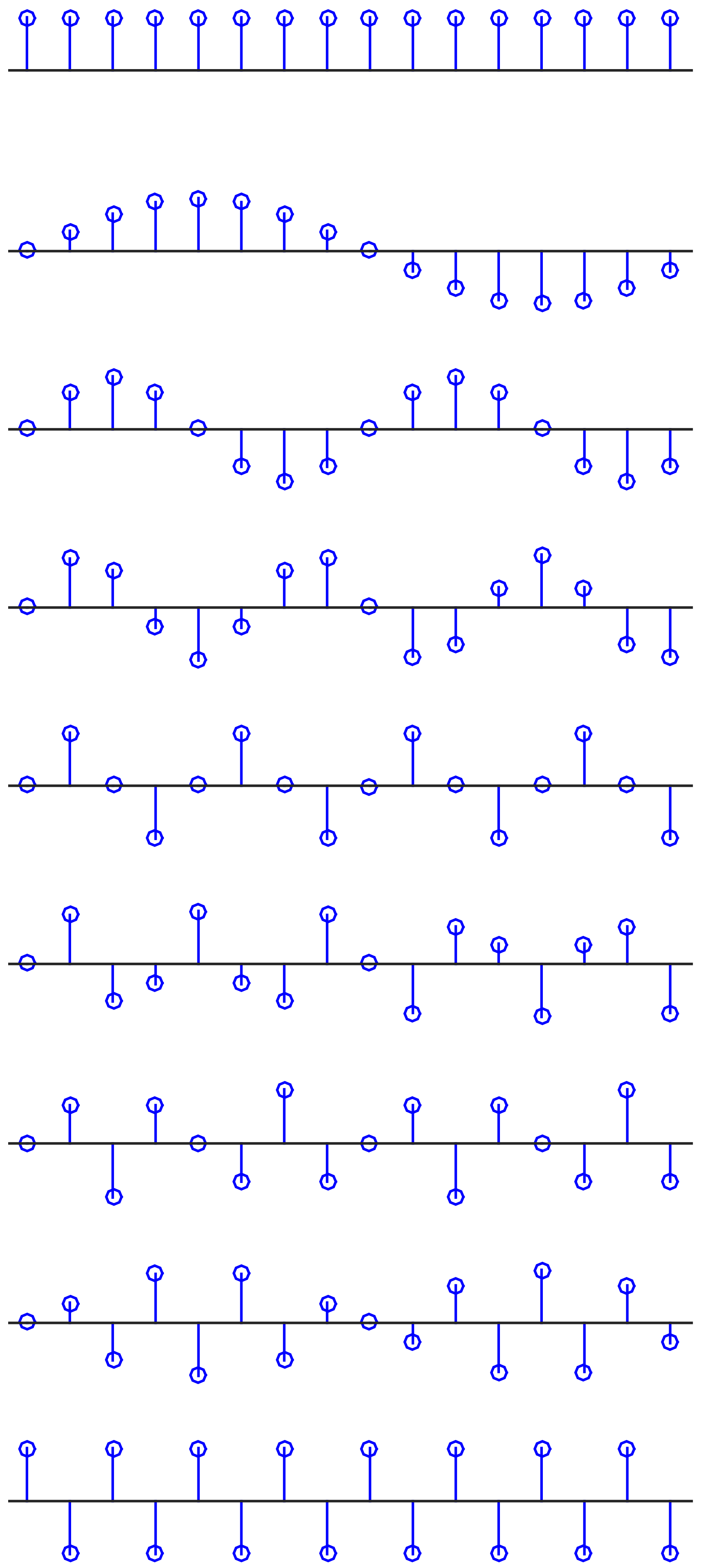
Filters and the Fourier Transform

$$y[t] = \frac{1}{2}x[t] - \frac{1}{2}x[t - \Delta t]$$



High Pass Filter

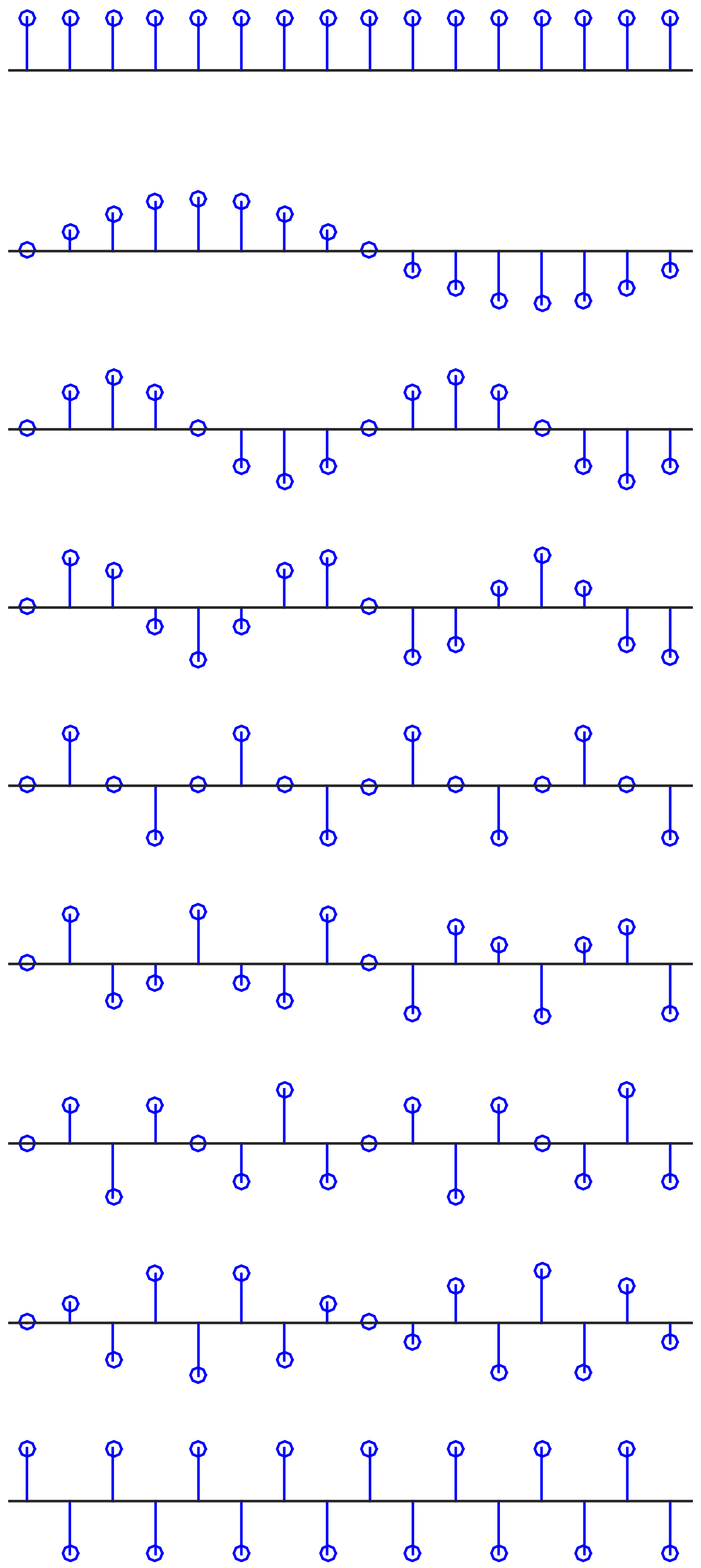
Filters and the Fourier Transform



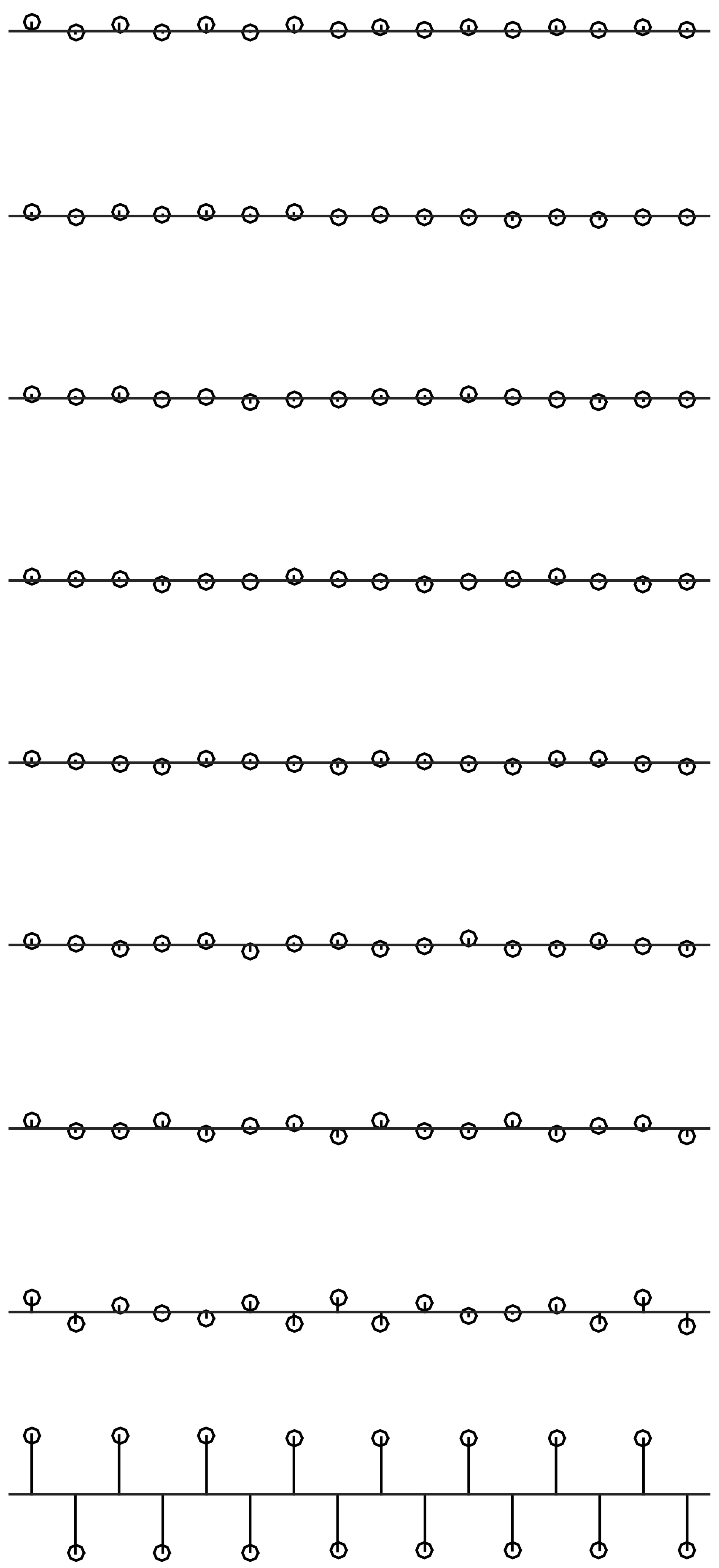
$$y[t] = \frac{1}{10}x[t] - \frac{9}{10}y[t-\Delta t]$$

?

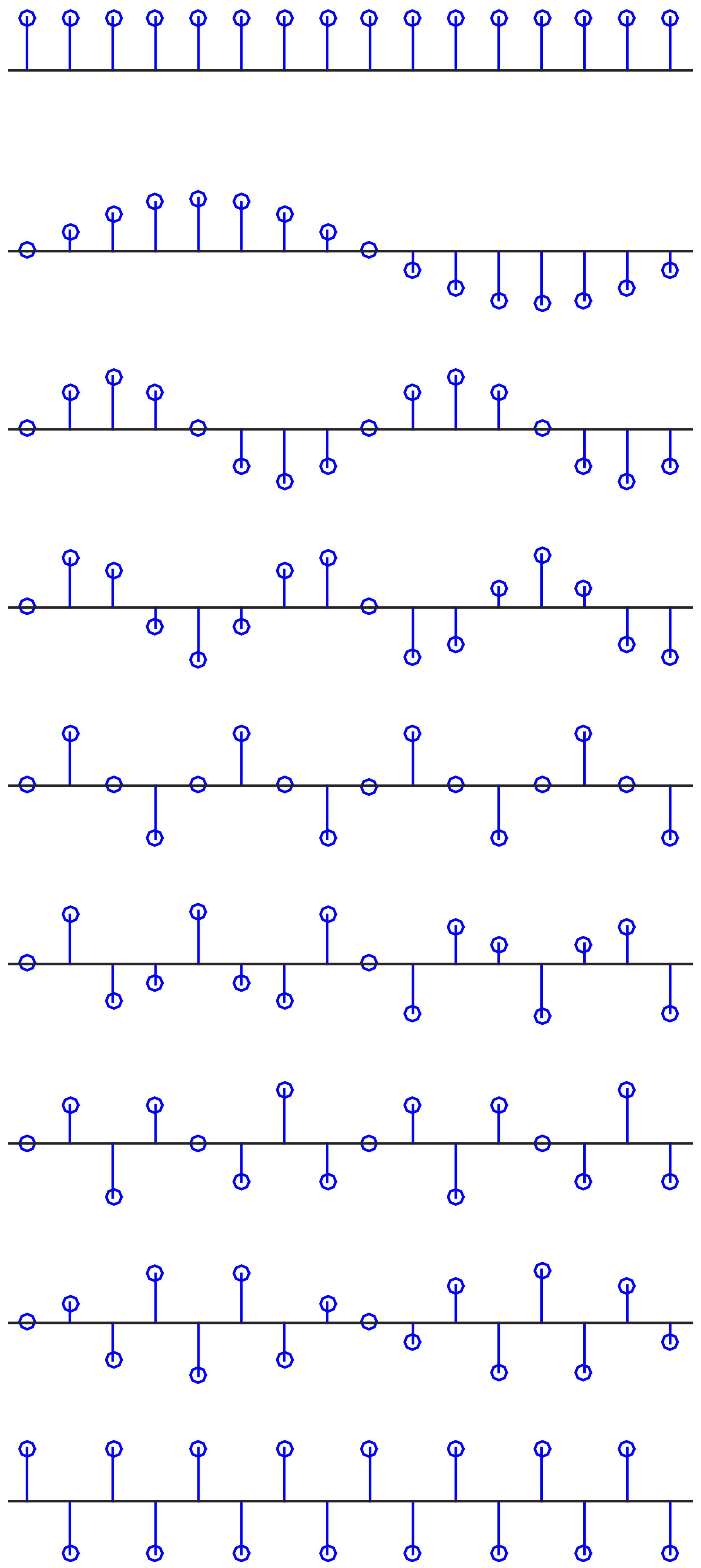
Filters and the Fourier Transform



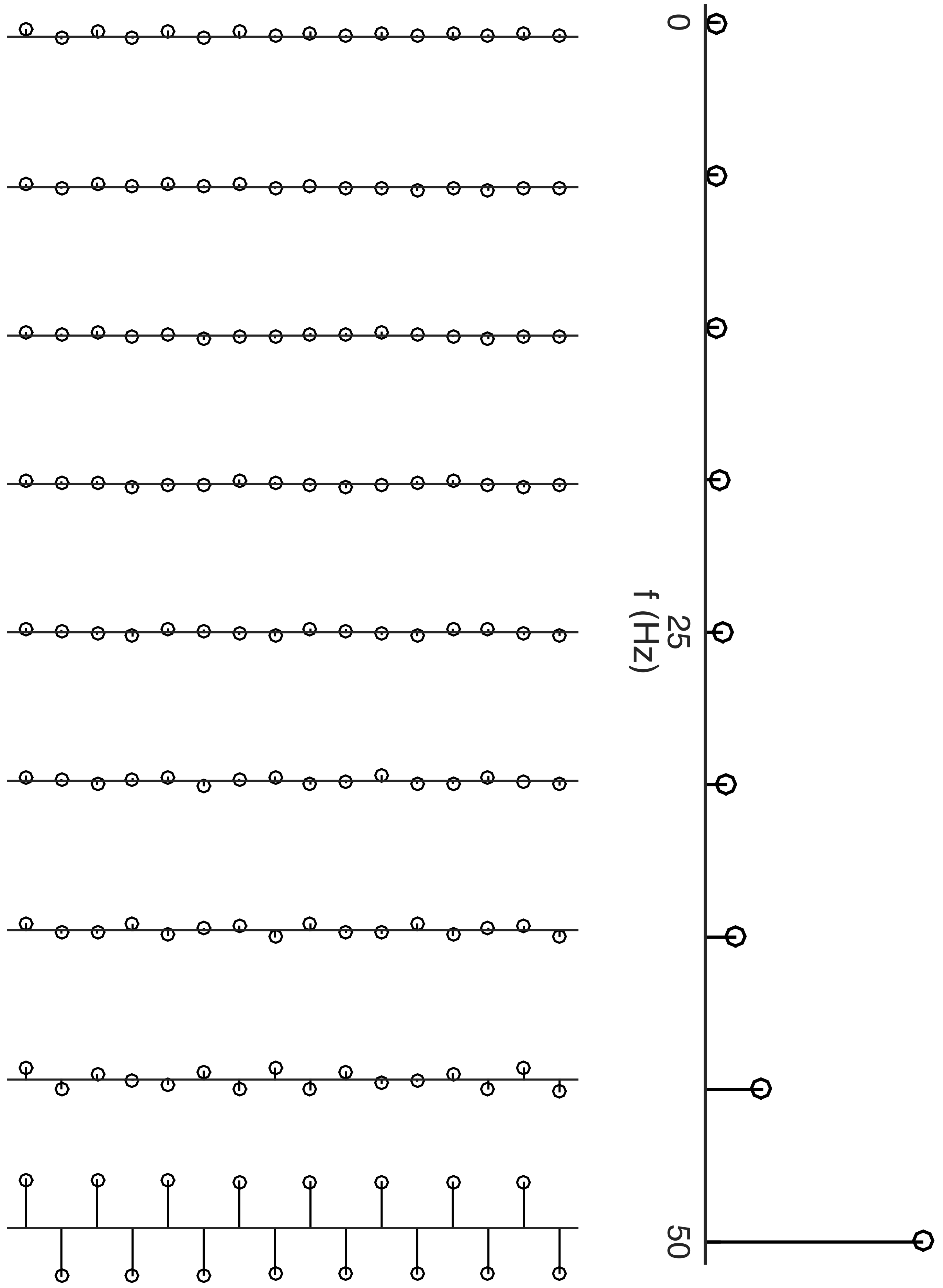
$$y[t] = \frac{1}{10}x[t] - \frac{9}{10}y[t - \Delta t]$$



Filters and the Fourier Transform

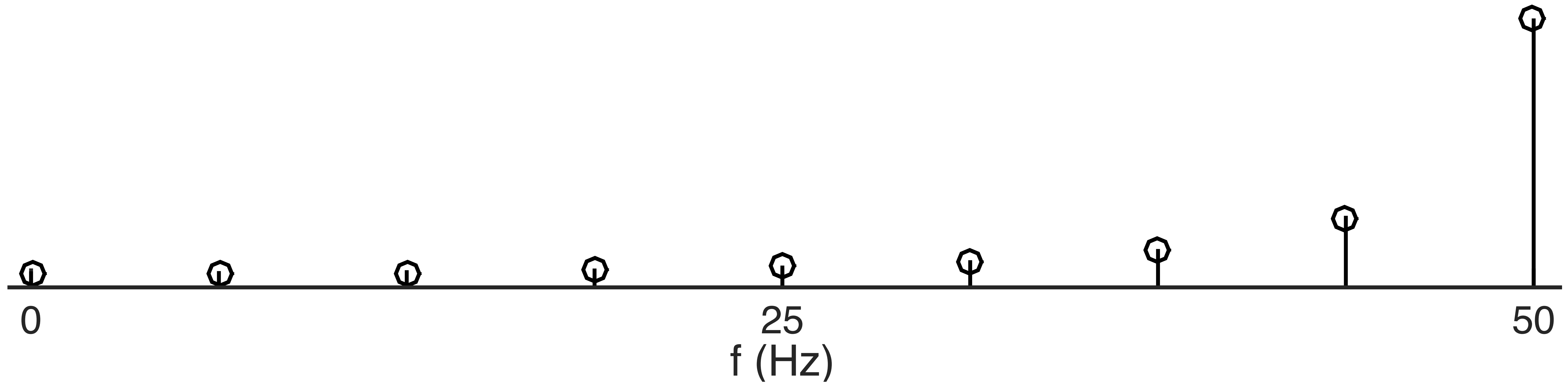


$$y[t] = \frac{1}{10}x[t] - \frac{9}{10}y[t-\Delta t]$$



Filters and the Fourier Transform

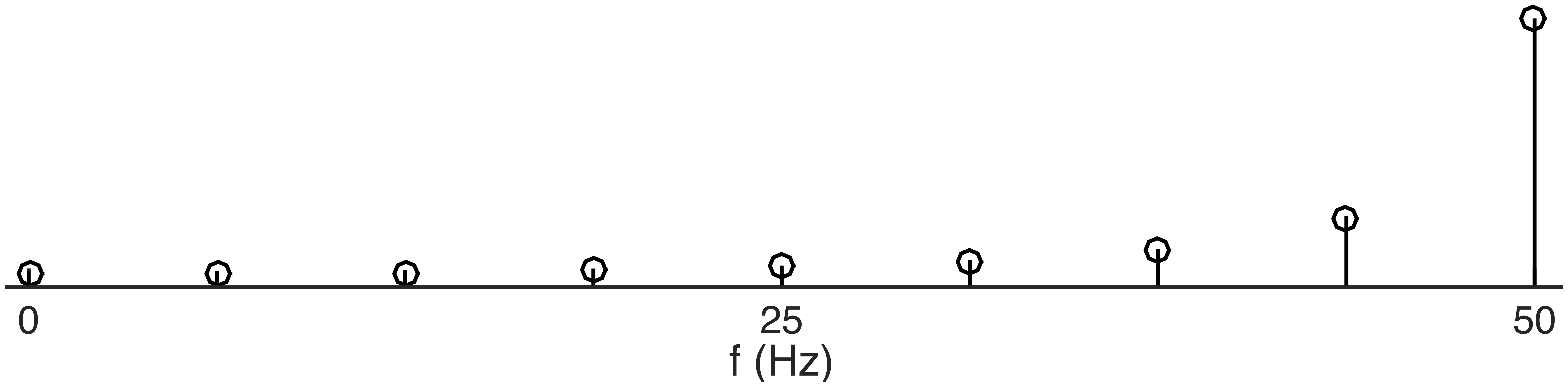
$$y[t] = \frac{1}{10}x[t] - \frac{9}{10}y[t-\Delta t]$$



High Pass Filter

Break for Computer Lab Exercise 4

$$y[t] = \frac{1}{10}x[t] - \frac{9}{10}y[t - \Delta t]$$



High Pass Filter

Outline

- Fourier Transform: *Why It's Useful*, and *What it Can/Cannot Do For You*
- Filters: *What They Do*, and *How They Do It*
- Filters: *Why So Many Different Kinds? Which Should I Use and When*?
- Grab Bag:
  - *Use Causal Filters; Windowing is Good*

Outline

- Fourier Transform: *Why It's Useful*, and *What it Can/Cannot Do For You*
- Filters: *What They Do*, and *How They Do It*
- Filters: *Why So Many Different Kinds? Which Should I Use and When*?
- Grab Bag:
  - *Use Causal Filters; Windowing is Good*

“Which Filter Should I Use?”

–Almost every student I’ve ever worked with

Many Filter Decisions

- Frequency Selectivity: Sharp vs. Soft Frequency Transition
- Feedforward Only/Feedback: FIR vs. IIR
- Filter Order: Low order vs. High Order
- Causality: Causal vs. non-Causal (e.g. “zero-phase” filters)
- and more (e.g., FIR: moving average vs. Parks-McClellan, IIR: Butterworth vs. elliptic)

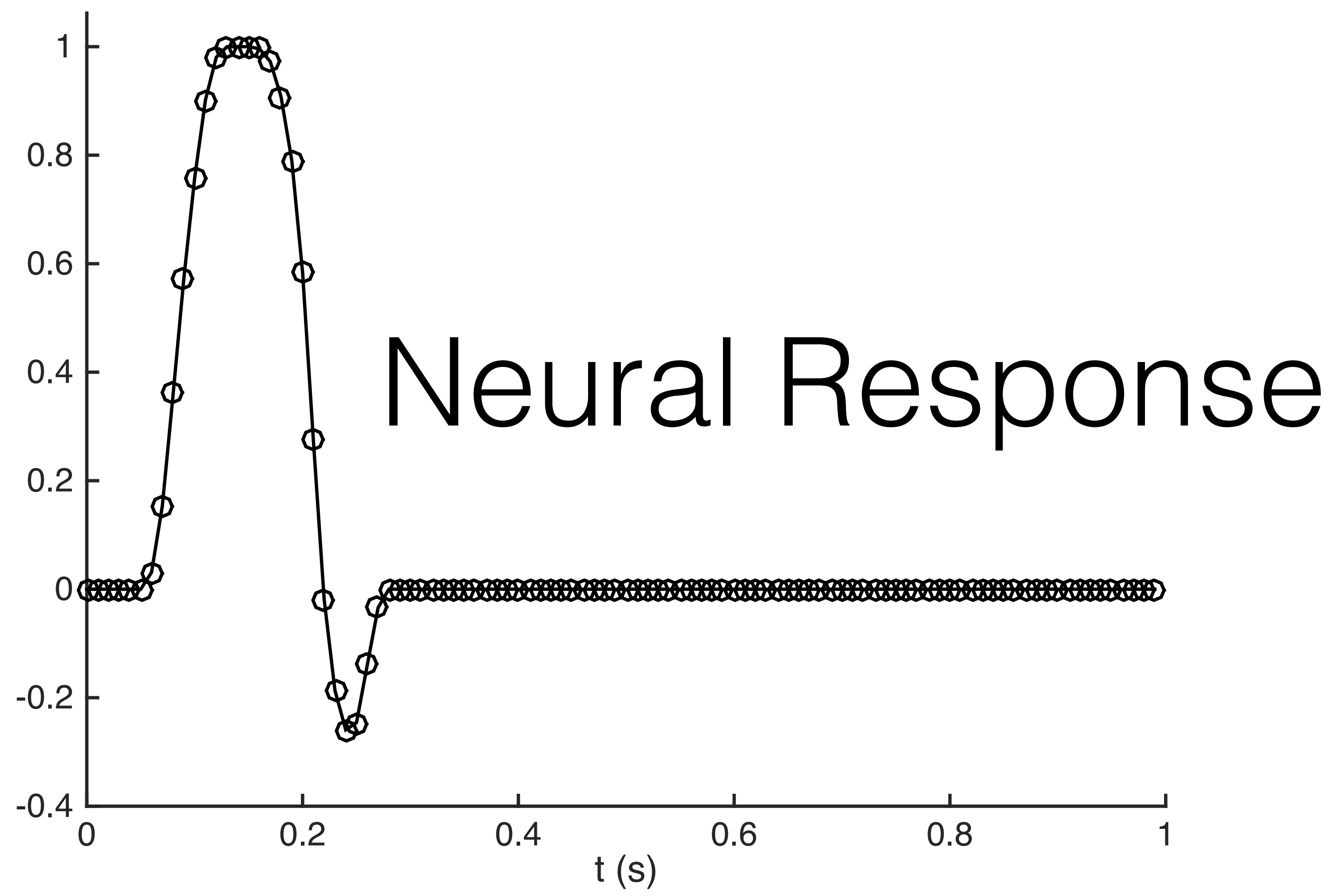
Ideas to Keep in Mind

- Filters modify signals, *by design.*
- There is no such thing as a filter that leaves signals (or signal components) unaltered
- Most filter decisions involve considering valid tradeoffs
  - Don’t go overboard one way or the other (if you do, be prepared).
- *Some* filter decisions allow us to avoid artifacts *without* any tradeoff

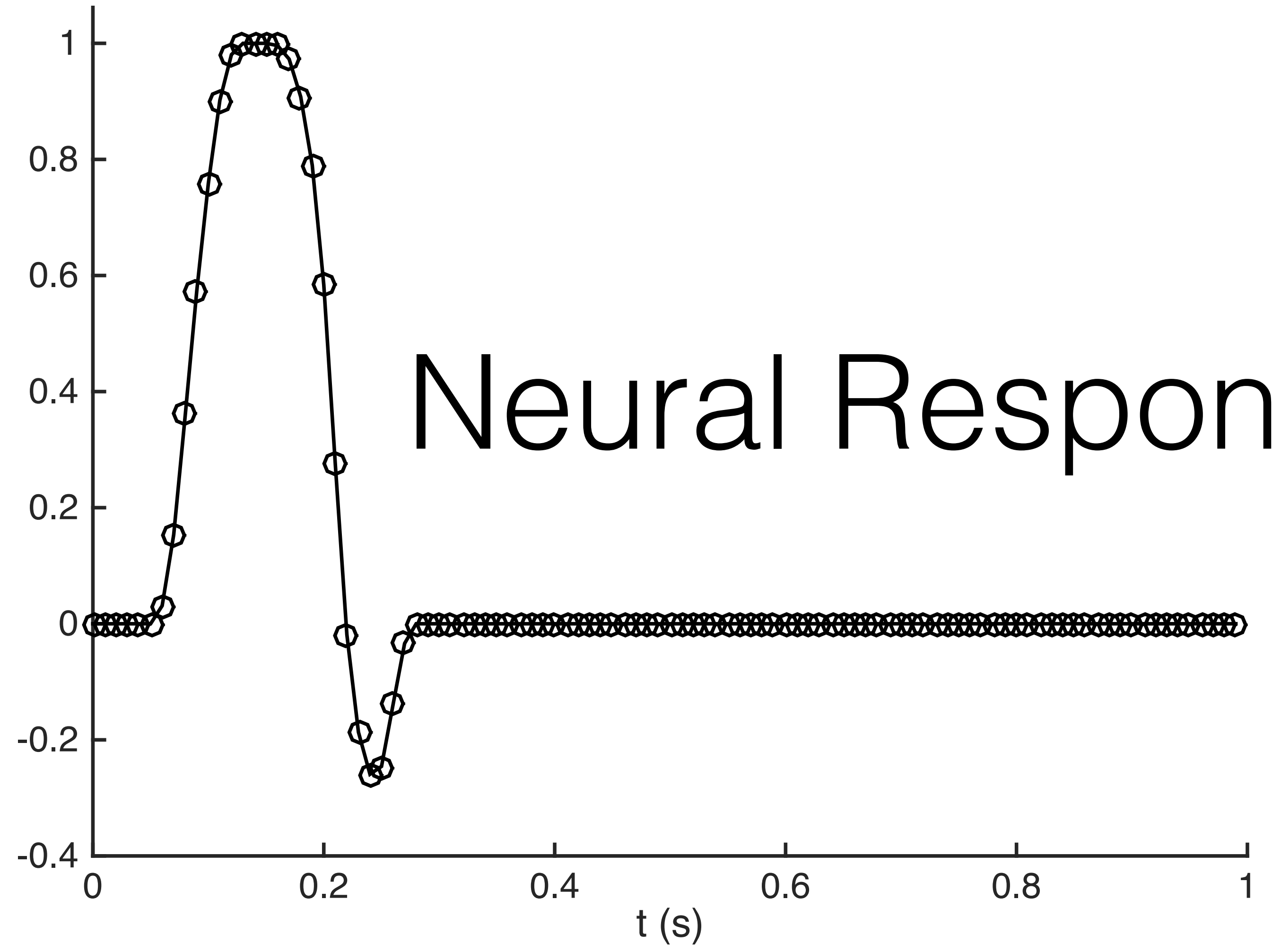
Frequency Selectivity/Transitions

- Time and Frequency are inextricably linked.
  - Changing the frequency content of a signal will change the temporal content of the signal.
    - Low-Pass Filters will lengthen fast temporal changes
    - High-Pass Filters will remove slow transitions from one baseline to another
  - Sharp frequency transitions produce artificial temporal elongation: “ringing”.

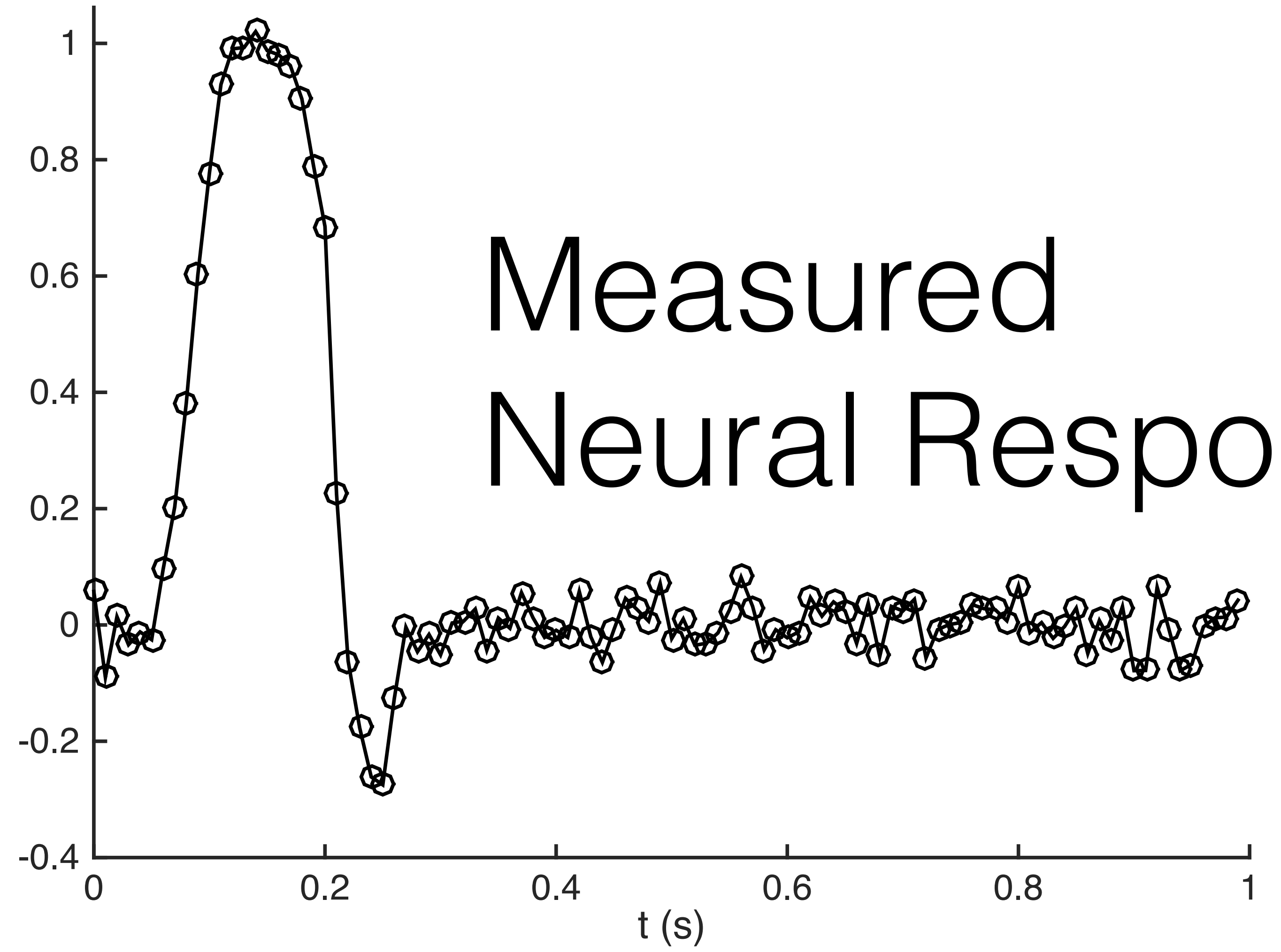
Ringing Artifacts



Ringing Artifacts

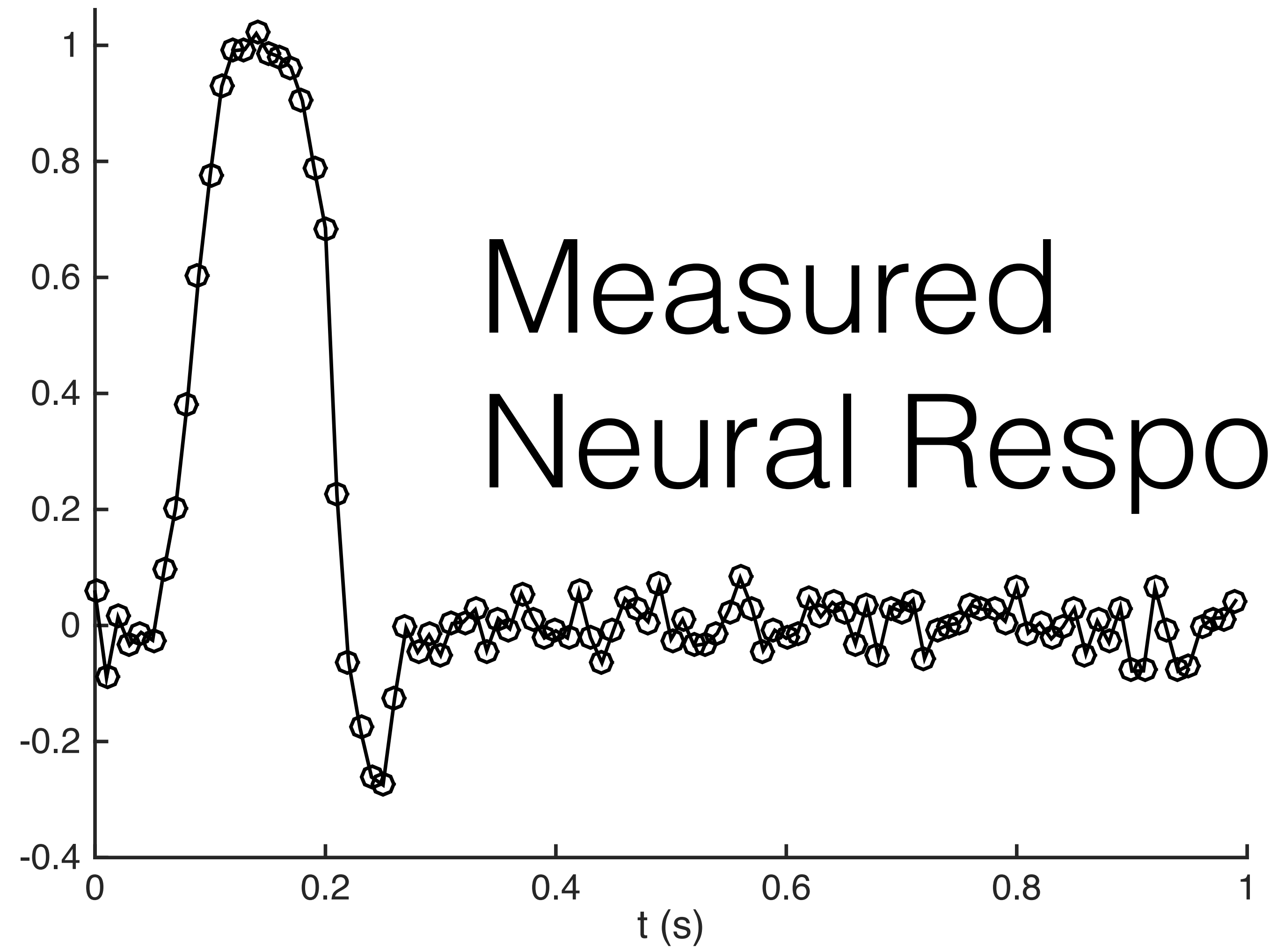


Neural Response



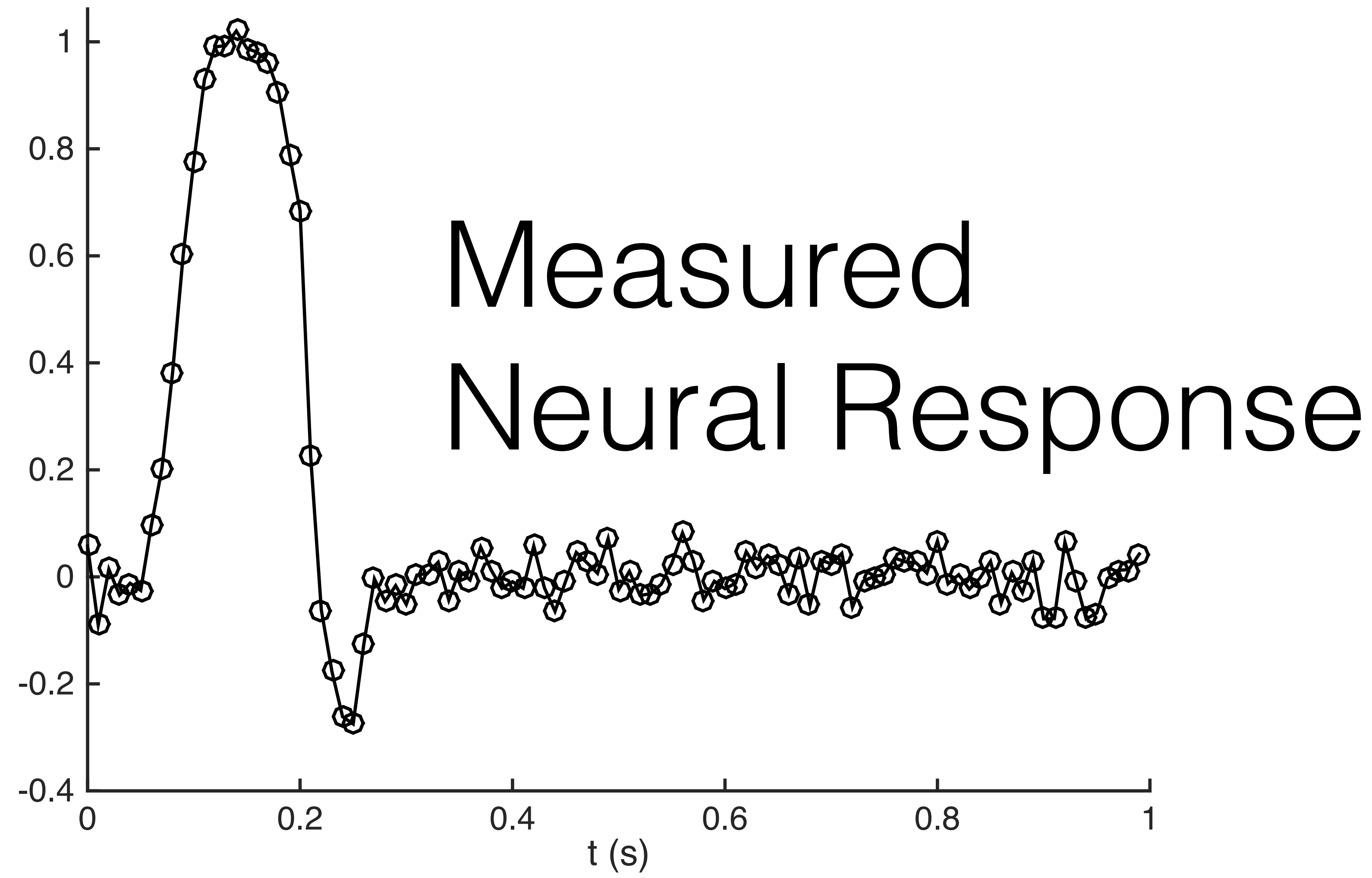
Measured
Neural Response

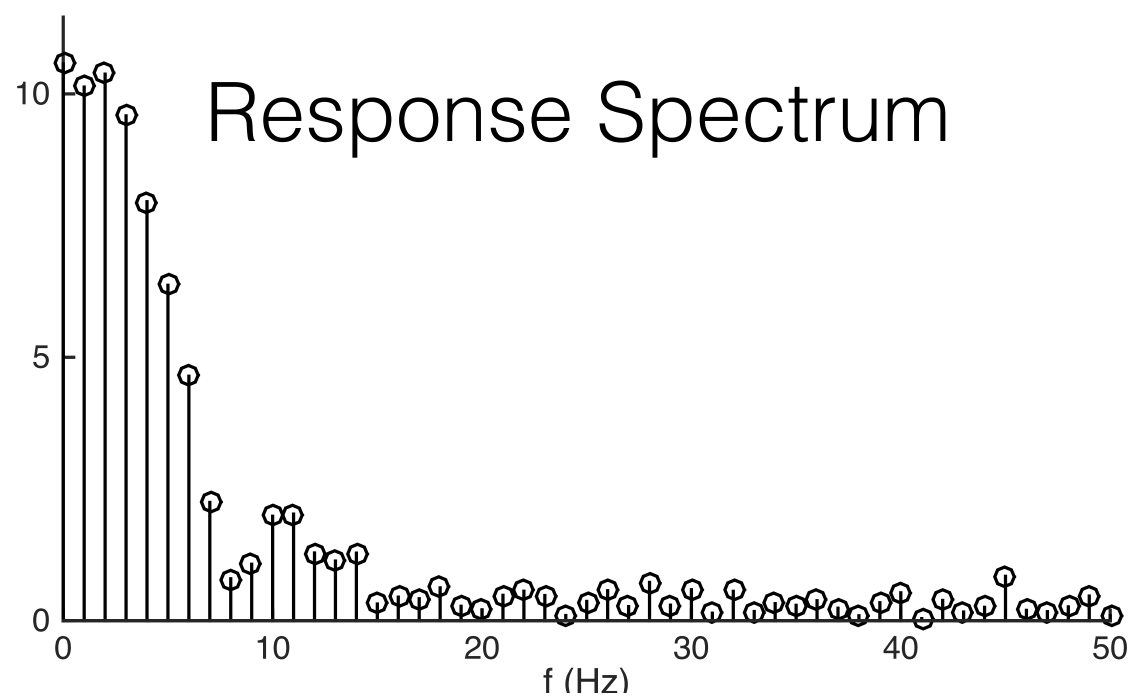
Ringing Artifacts



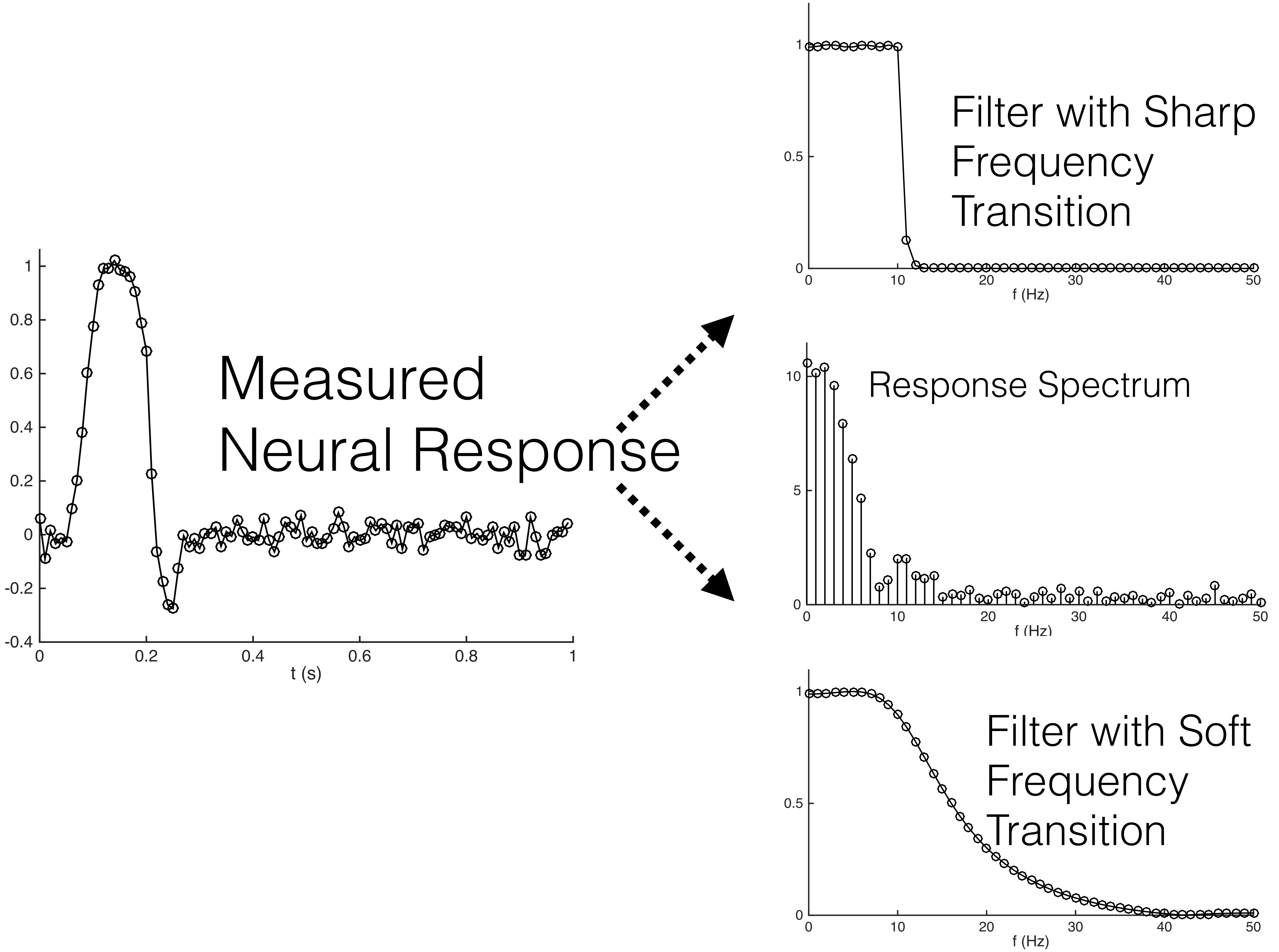
Measured
Neural Response

Ringing Artifacts

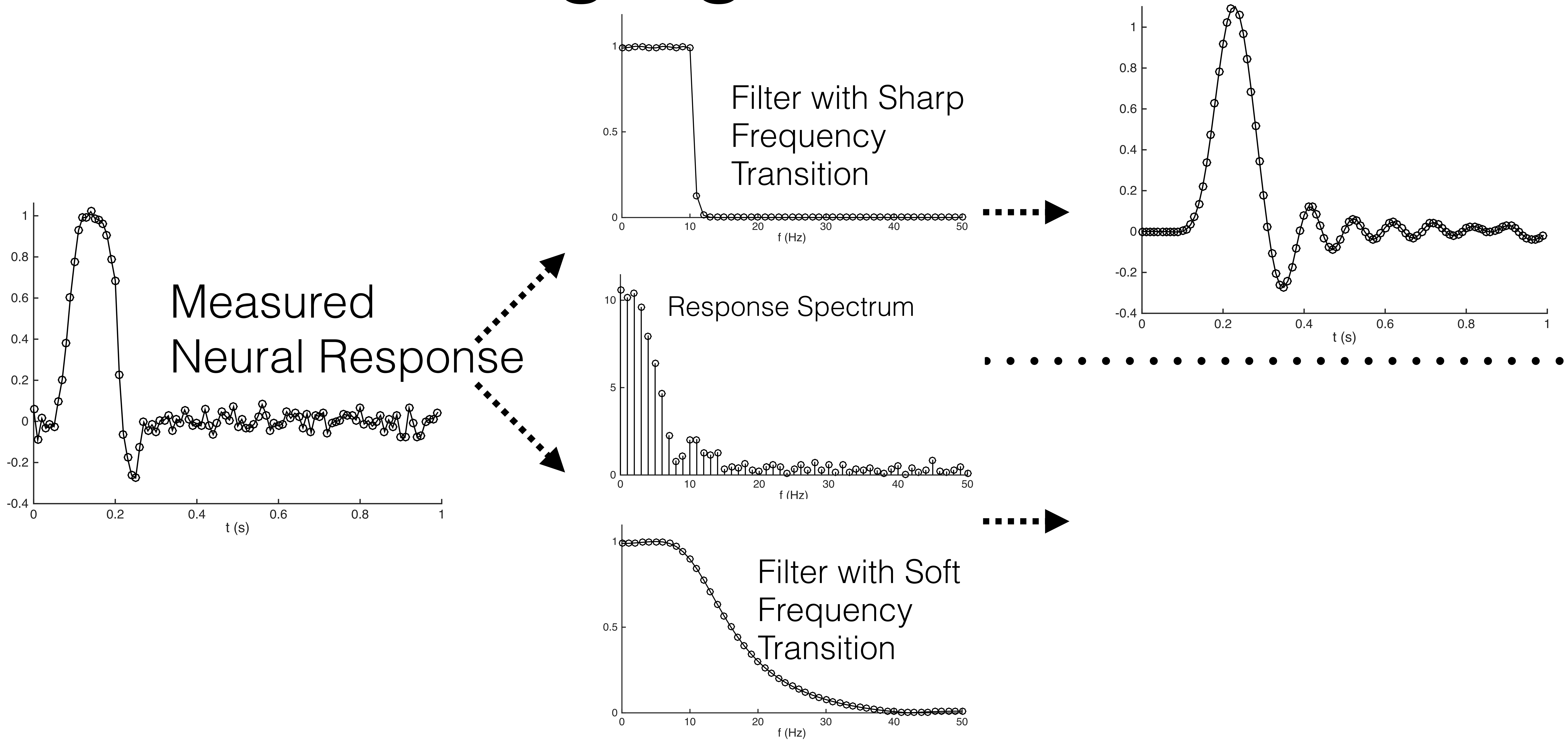




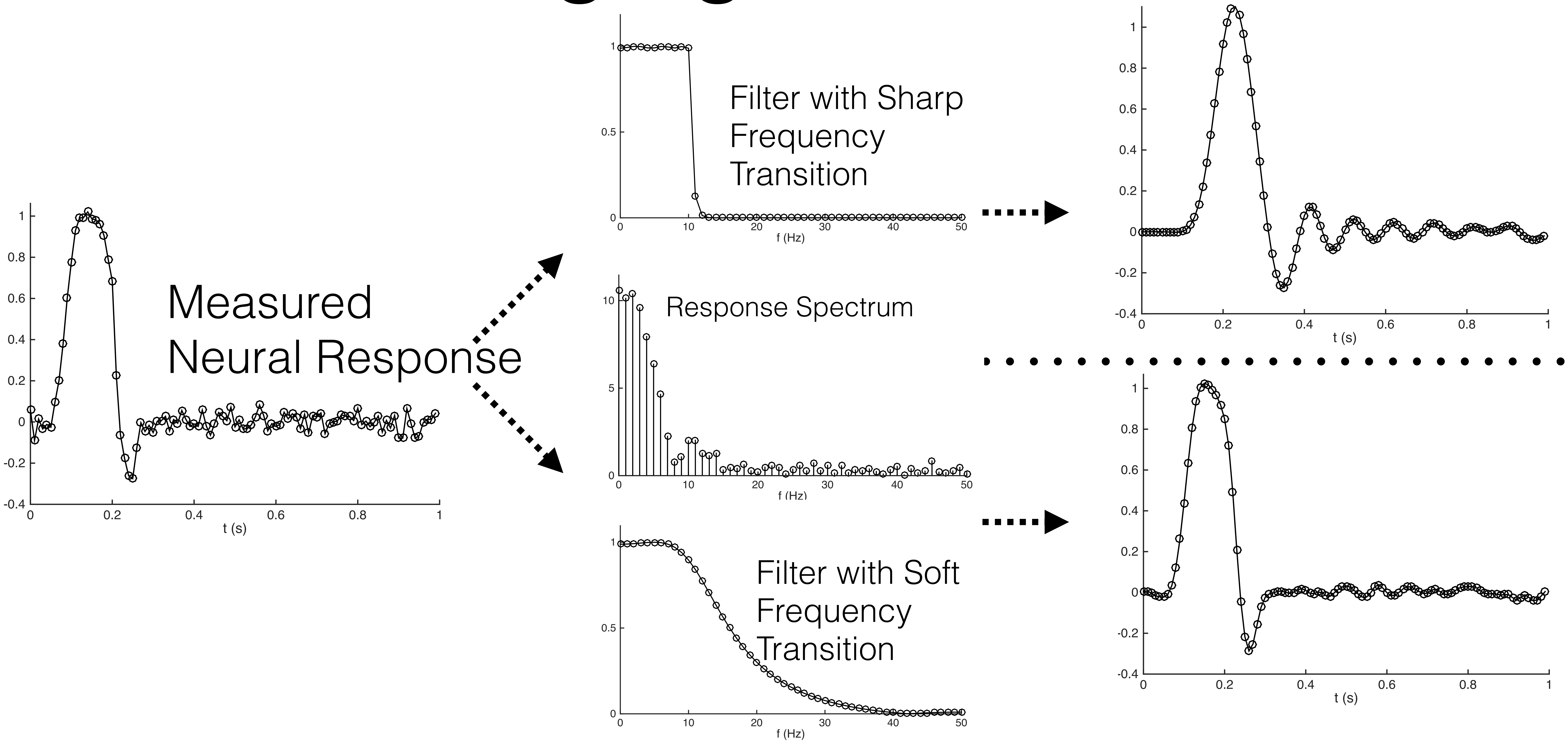
Ringing Artifacts



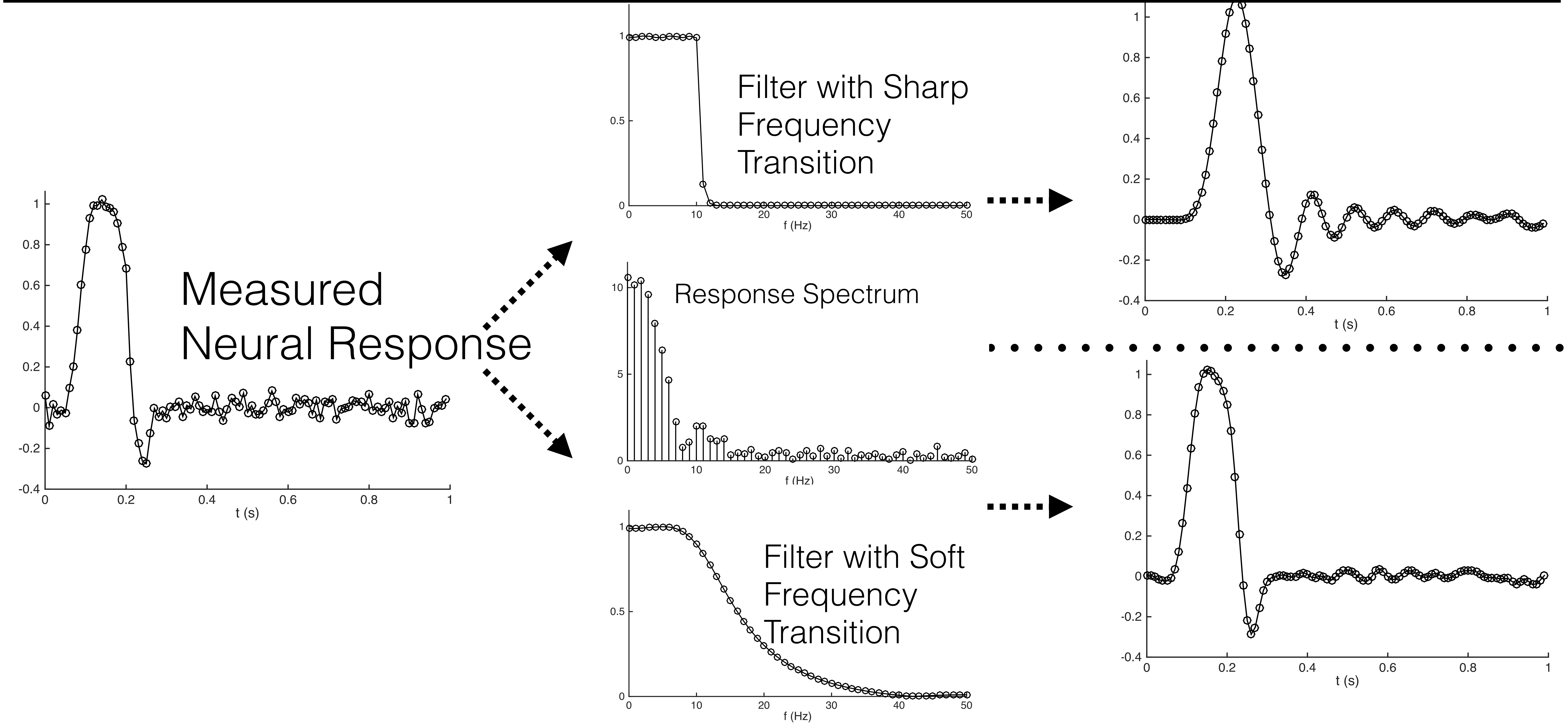
Ringing Artifacts



Ringing Artifacts



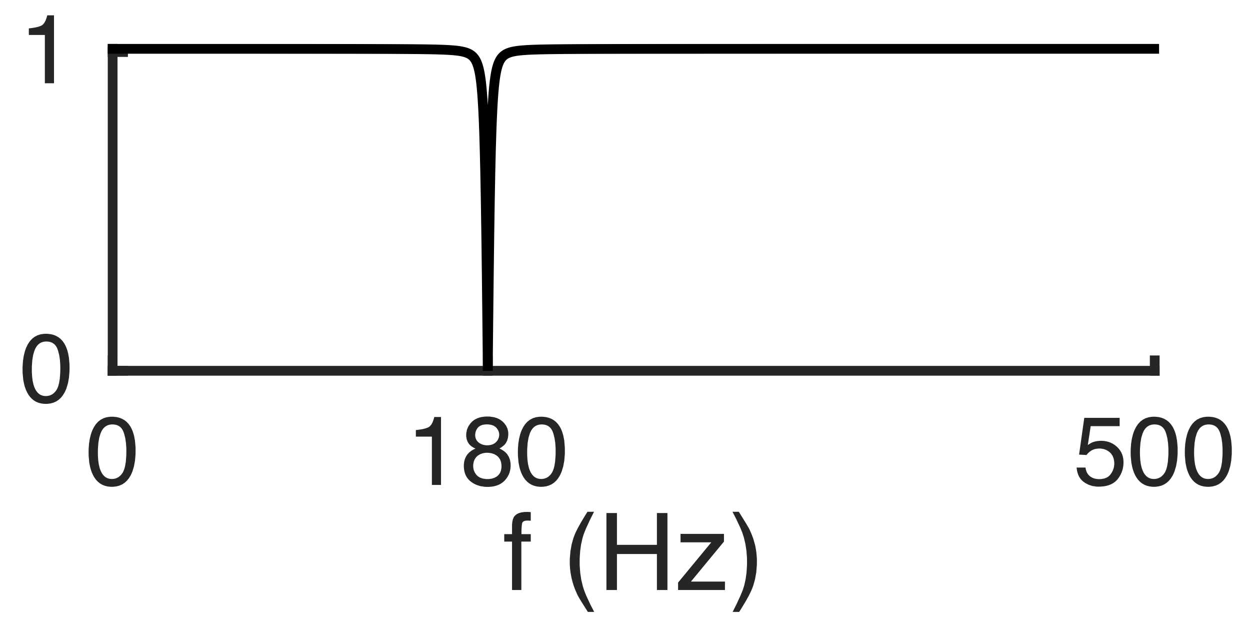
Break for Computer Lab Exercise 5



Ringing Artifacts

- Sharp Frequency Transitions are sometimes Necessary
  - e.g., Notch filters (and related filters, such as Comb filters)
- In these cases there will be unavoidable ringing

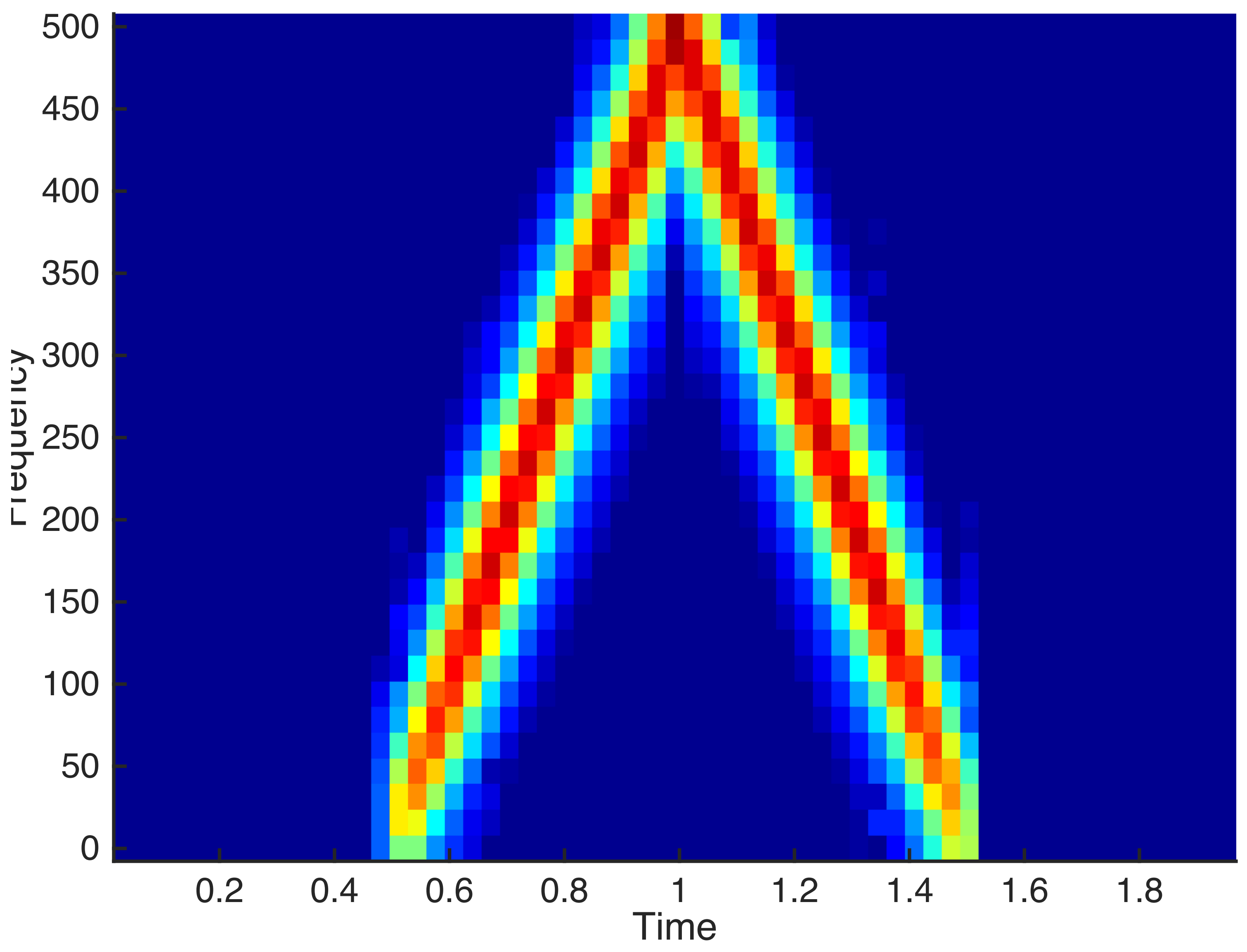
Ringing Artifacts

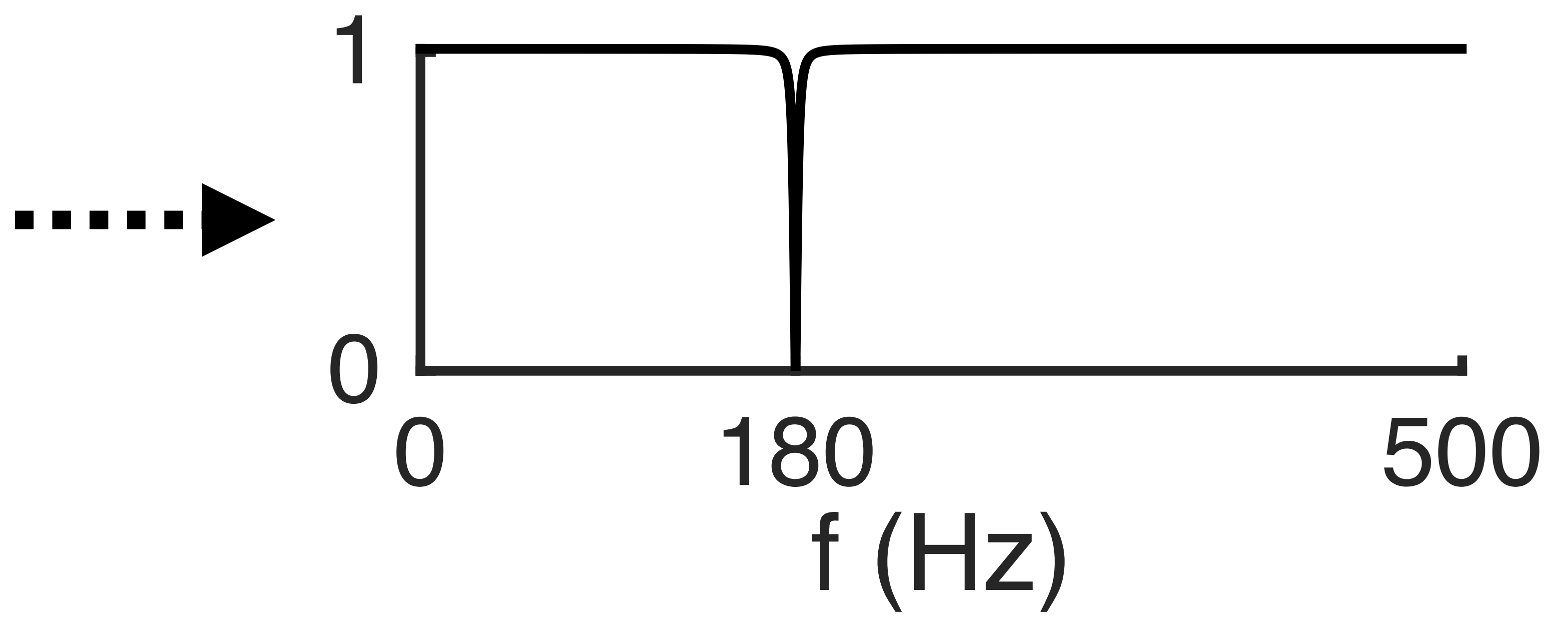


Notch Filter
(Sharp Frequency Transition)

Ringing Artifacts

FM Sweep
(Spectrogram)

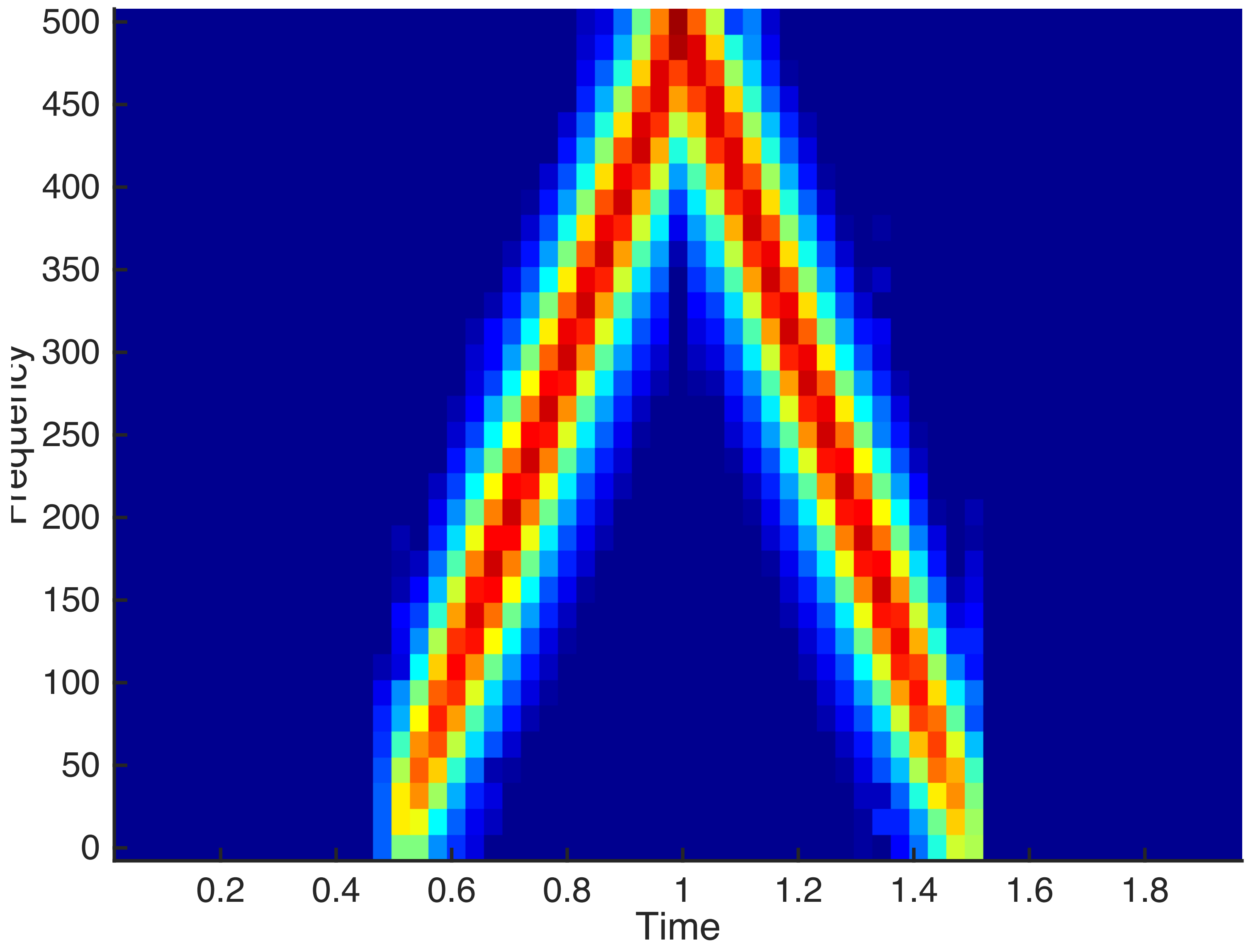


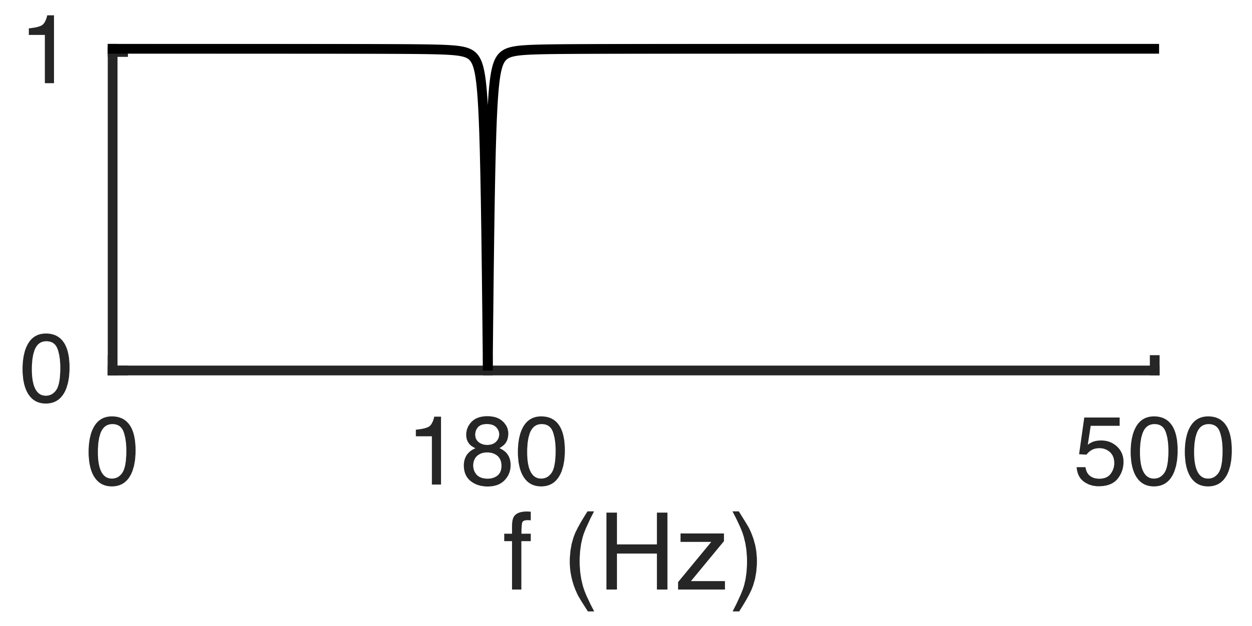


Notch Filter
(Sharp Frequency Transition)

Ringing Artifacts

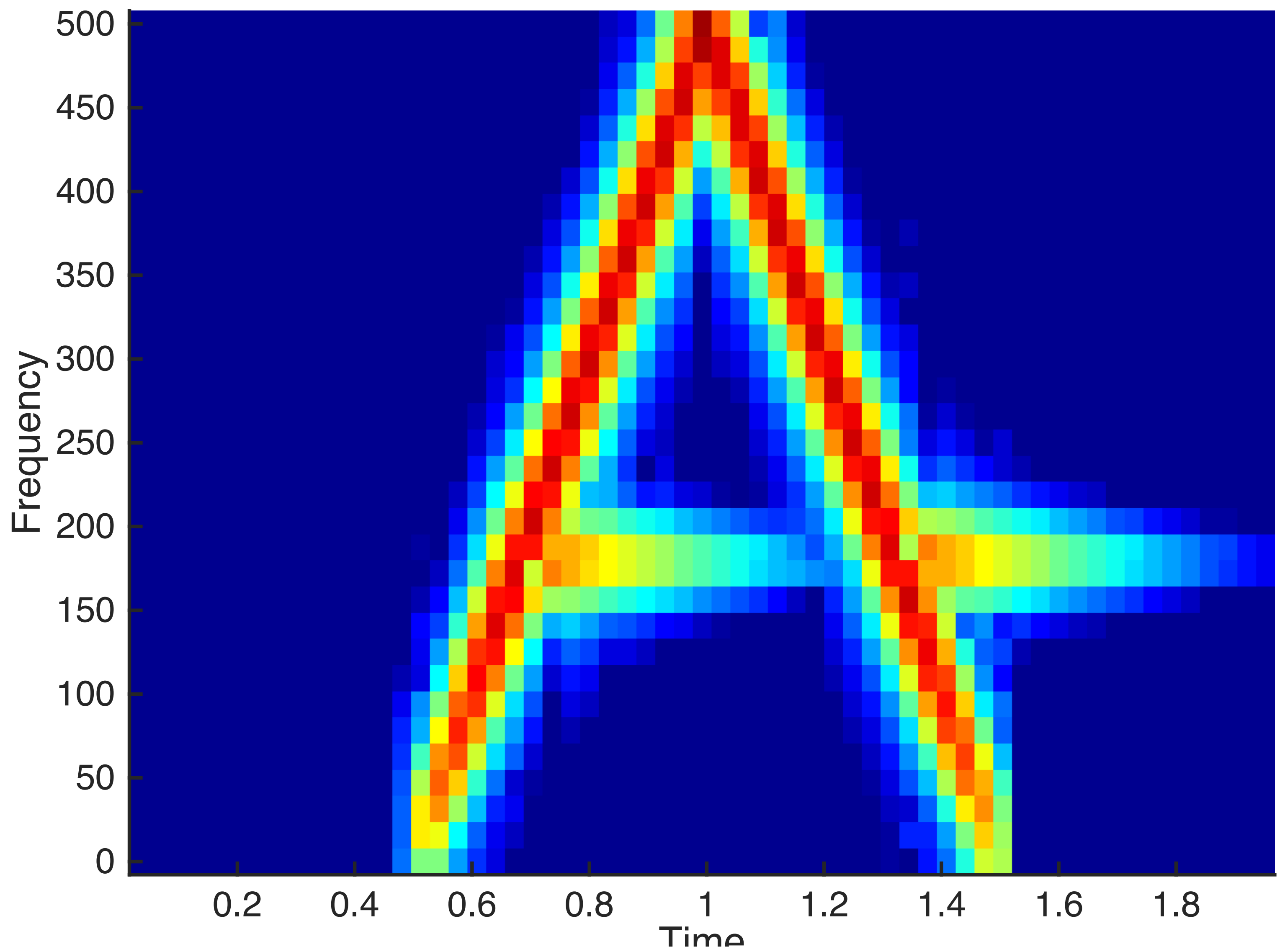
FM Sweep
(Spectrogram)





Notch Filter
(Sharp Frequency Transition)

Notched FM Sweep
(Spectrogram)

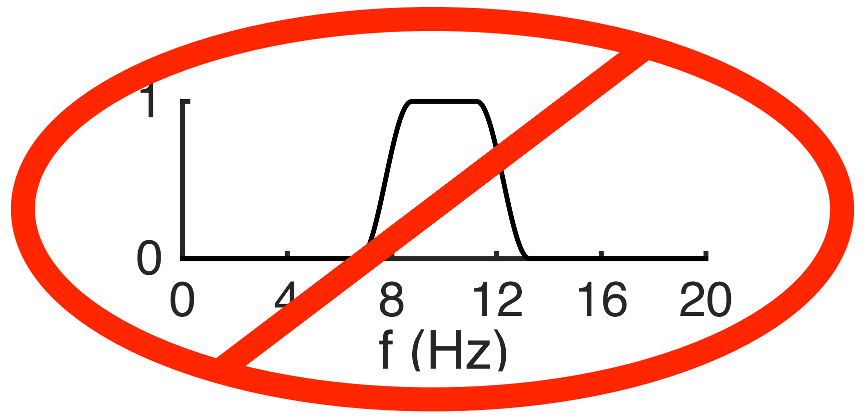


Notch too brief to see
But ringing clear:

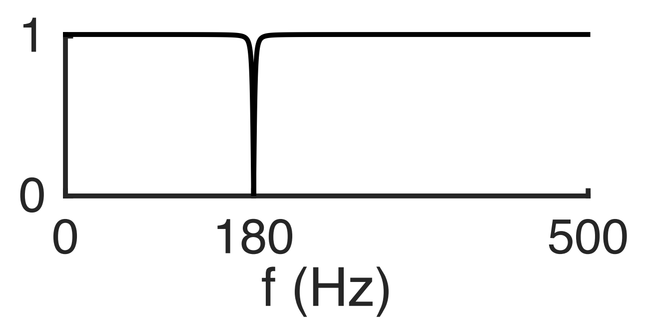
- narrowband
- extended in time

Take care, but don't overreact

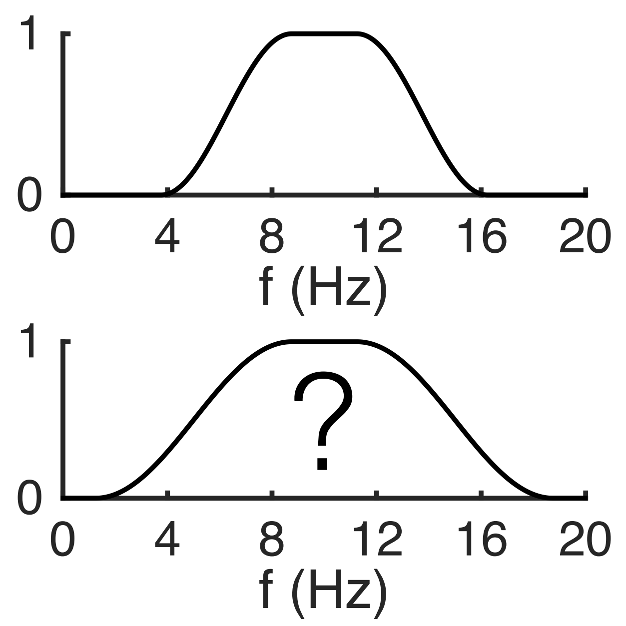
- Avoid Ringing by avoiding sharp frequency transitions



- If sharp frequency transitions are necessary (as for notch filtering), ringing may follow



- Don't overly soften frequency transitions or you'll lose frequency selectivity



FIR vs. IIR

- FIR (finite impulse response): Feedforward only
  - Examples: Moving Average (*avoid, in general*), Parks-McClellan ("Optimal"), others
- IIR (infinite impulse response): Feedback also incorporated
  - Instability a potential issue
  - Examples: Butterworth (*not awful, but not great*), Chebyshev, Elliptic (*very good*), others

FIR vs. IIR: How to choose?

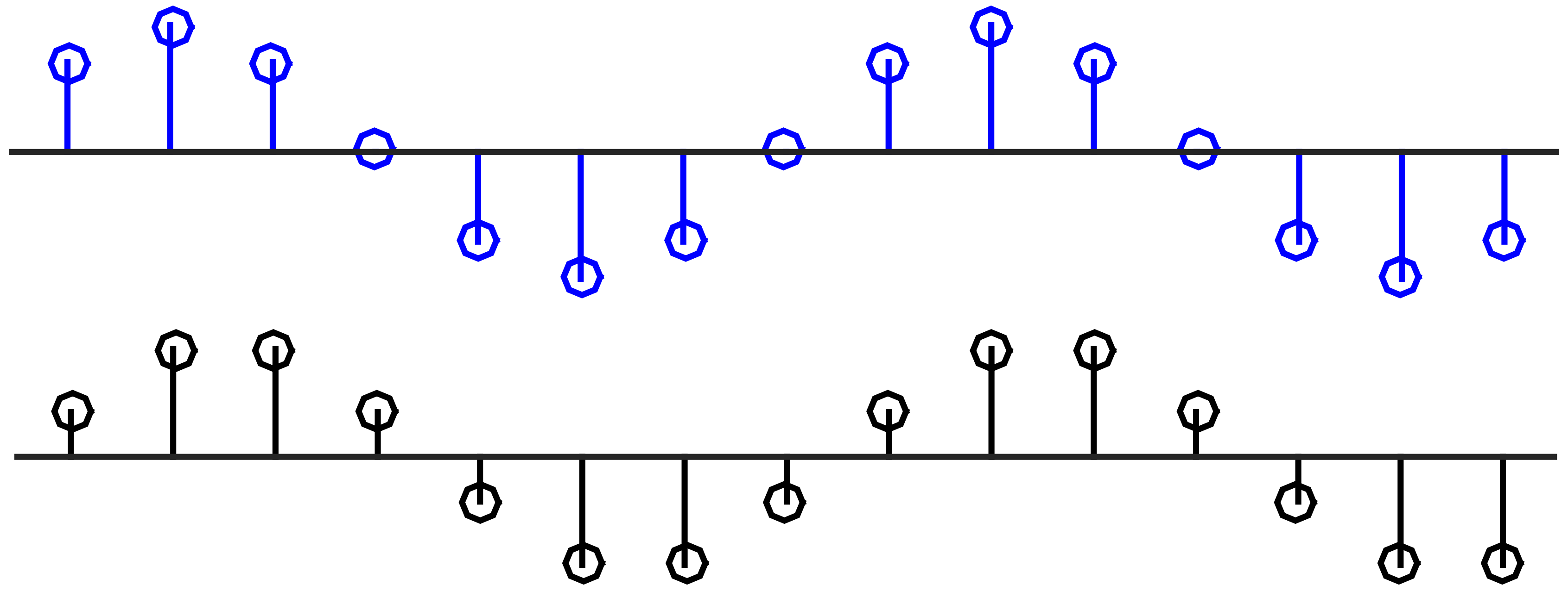
- No universal answer. It may depend on:
  - *group delay (*signal delay intrinsic to filter): both the value of the delay and frequency dependence of that value
  - signal loss due to filter startup (output value dependence on signal values before signal starts)
  - stability concerns (if IIR filter)
  - more…

Group Delay

- Intrinsic to filtering—cannot be removed
- Filtering changes signals by design—all filters change temporal features of the signal
- Causal filters always incur delay

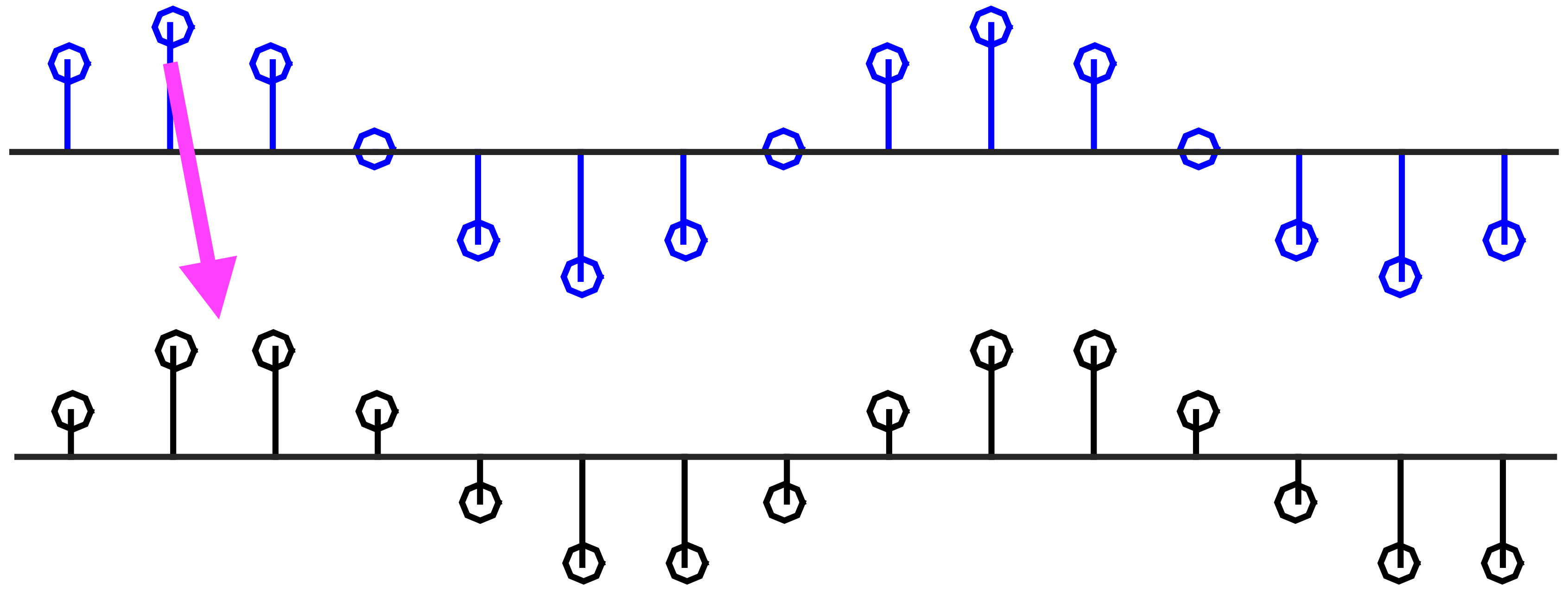
Group Delay Examples

$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t - \Delta t]$$



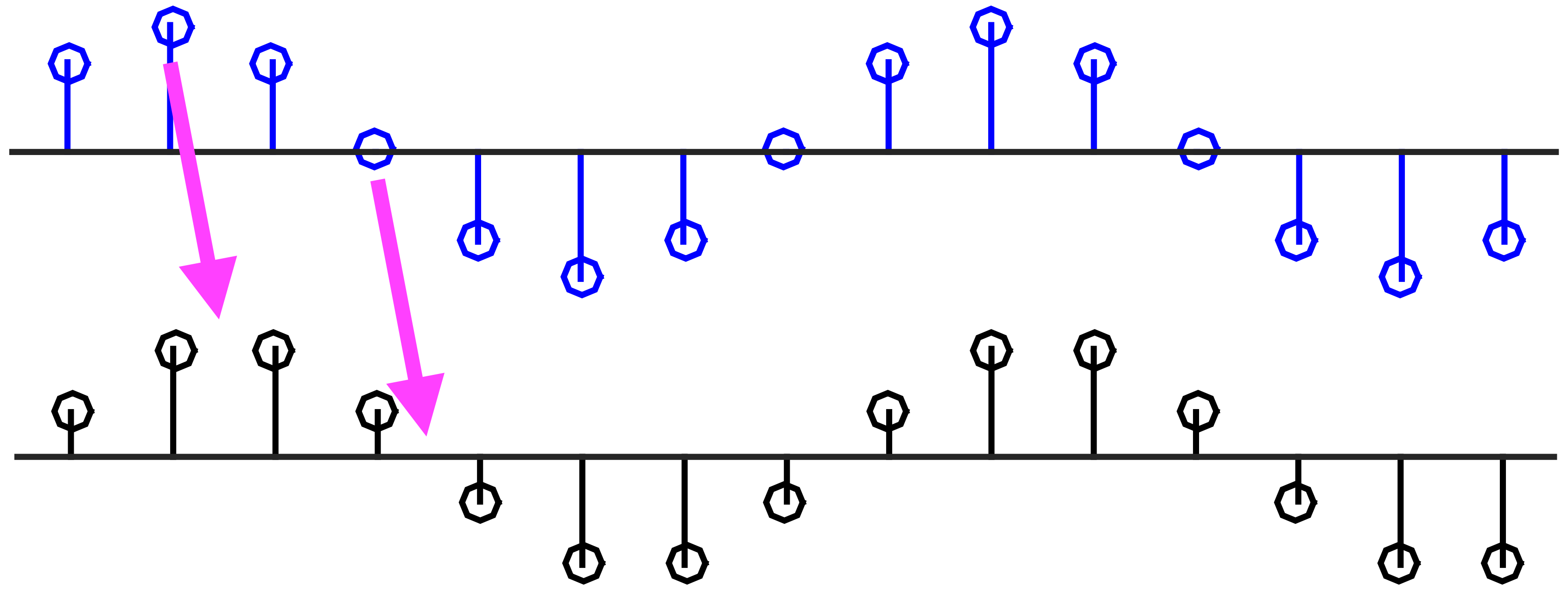
Group Delay Examples

$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t - \Delta t]$$



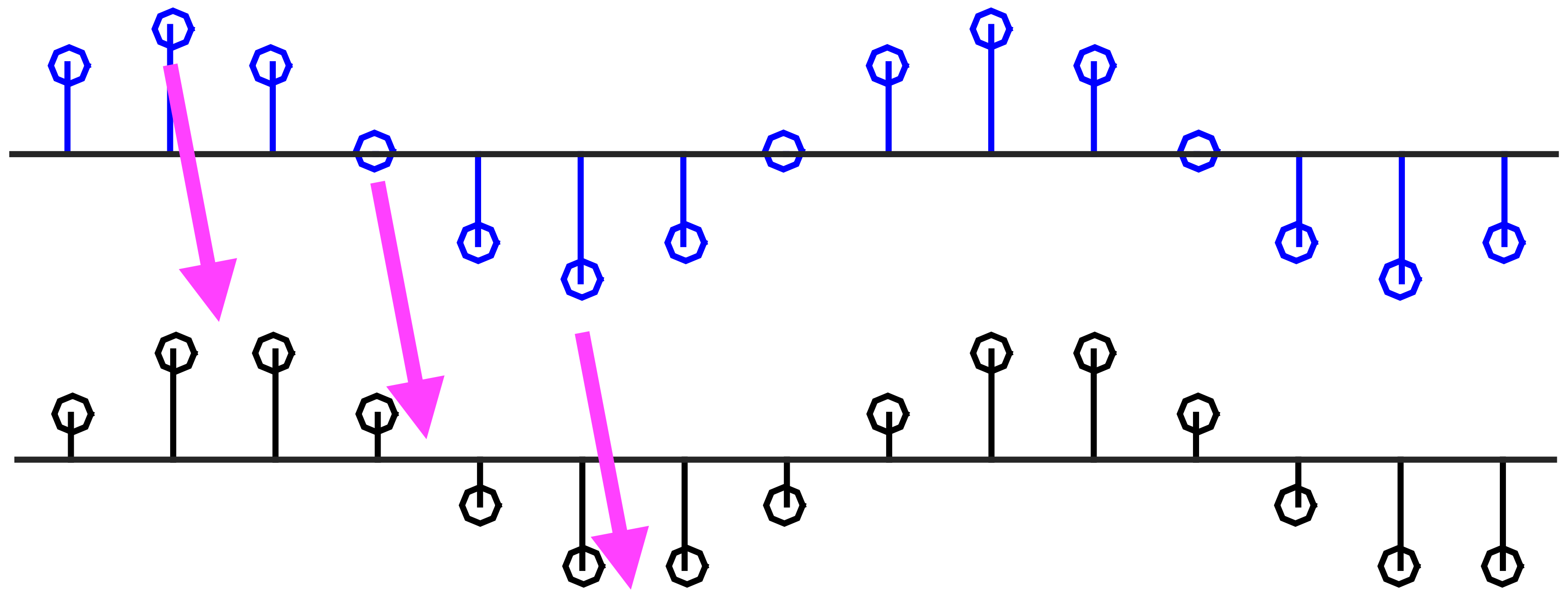
Group Delay Examples

$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t - \Delta t]$$



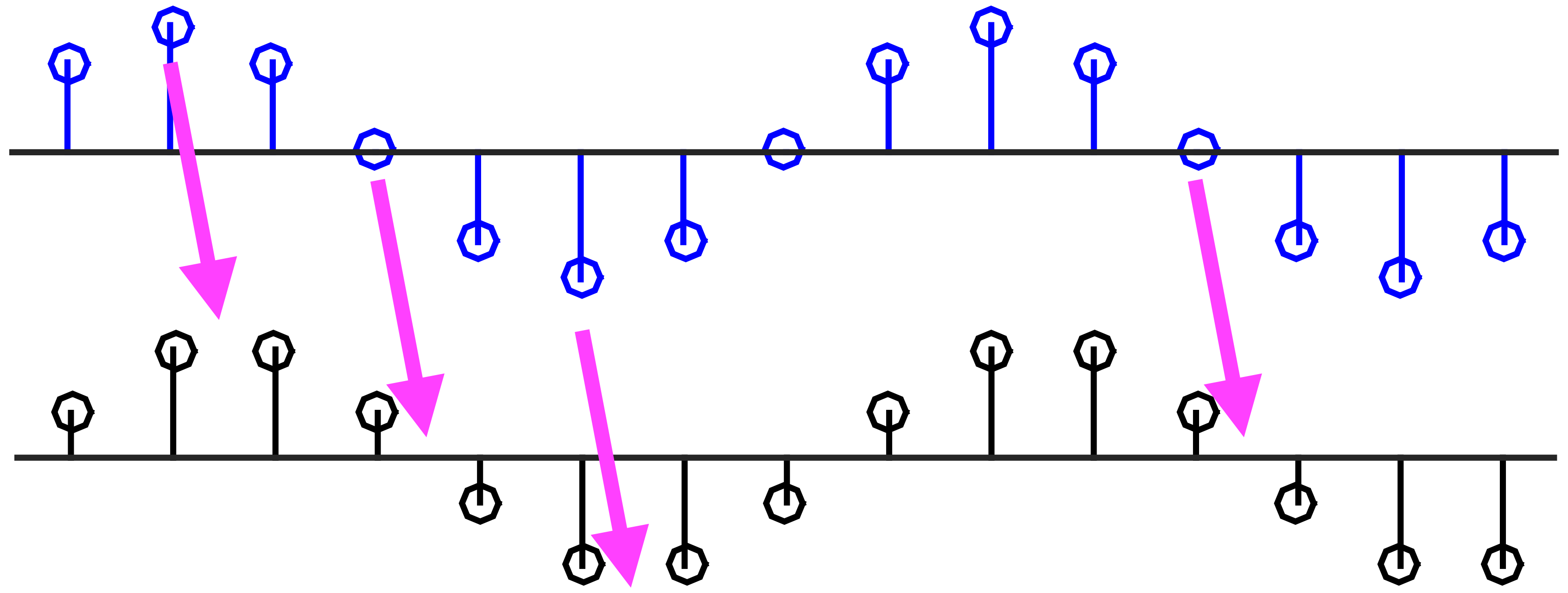
Group Delay Examples

$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t - \Delta t]$$



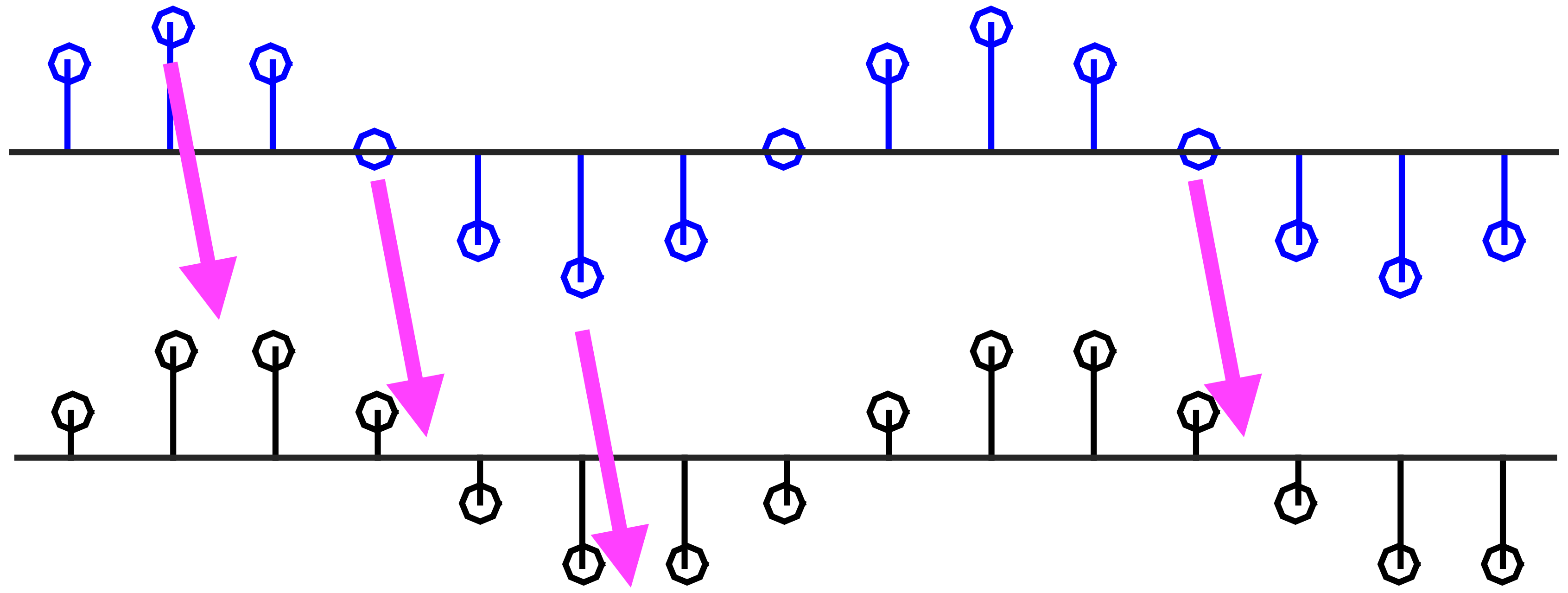
Group Delay Examples

$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t - \Delta t]$$



Group Delay Examples

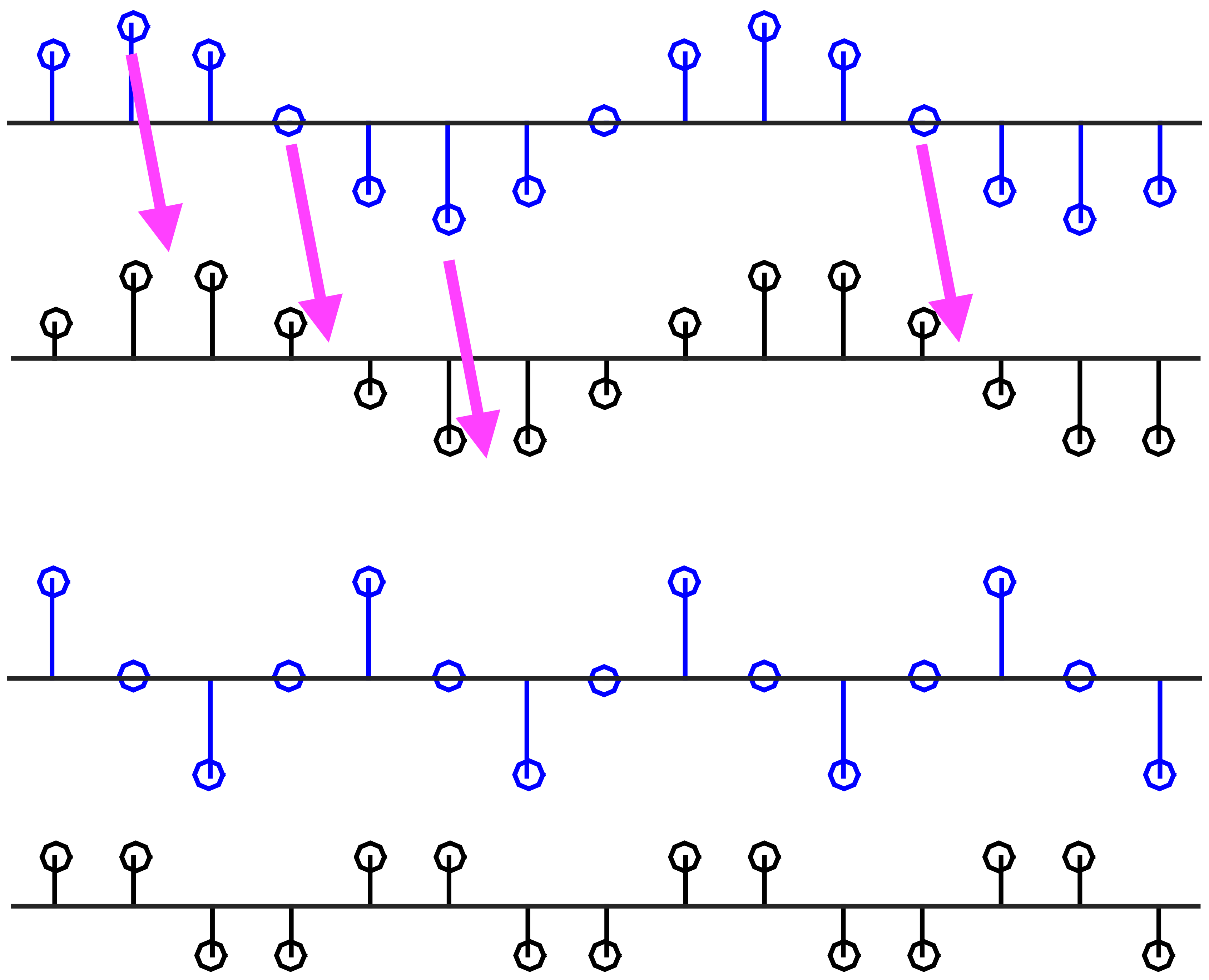
$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t - \Delta t]$$



τ_D: $\frac{\Delta t}{2}$(?)

Group Delay Examples

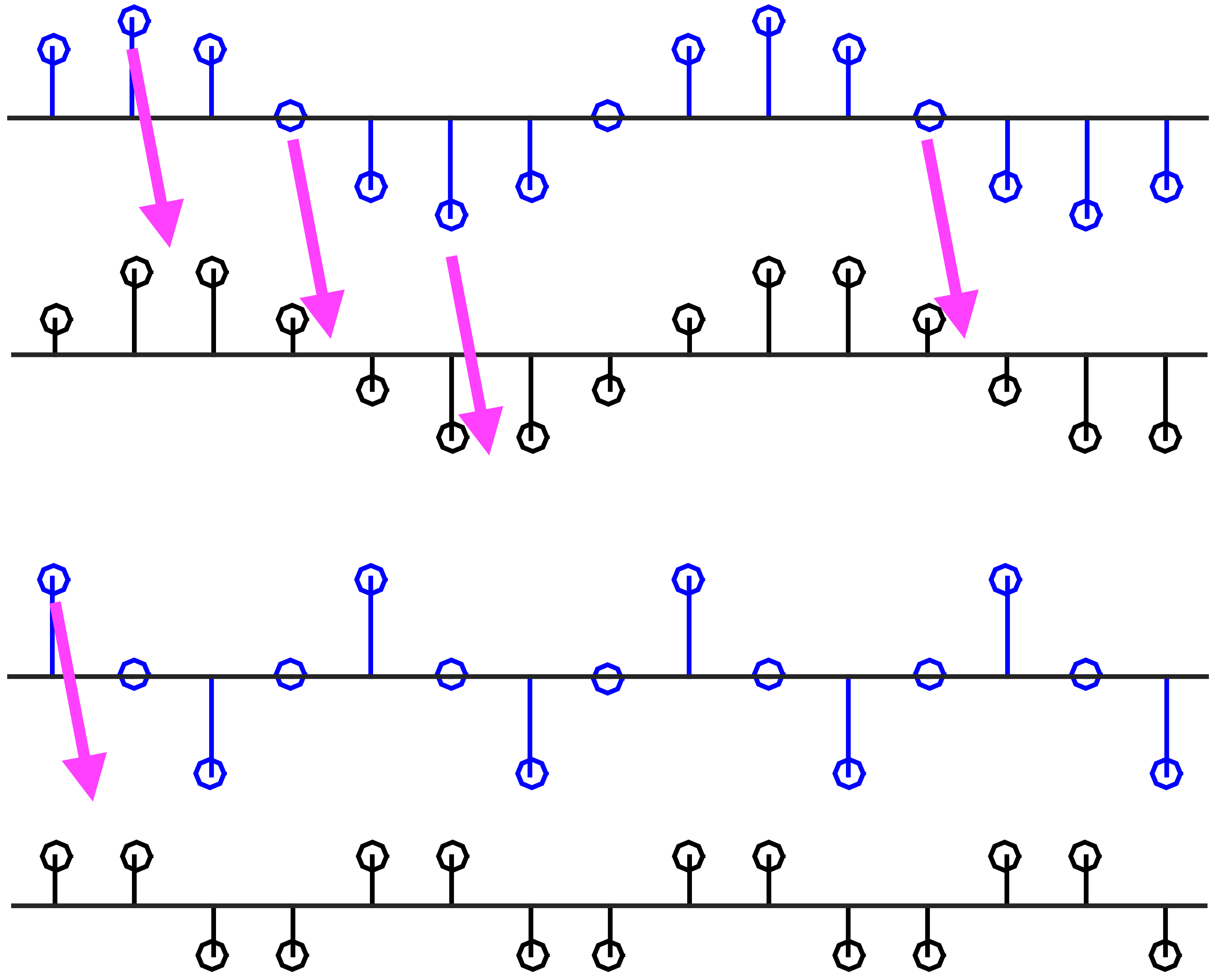
$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t - \Delta t]$$



τ_D: $\frac{\Delta t}{2}$ (?)

Group Delay Examples

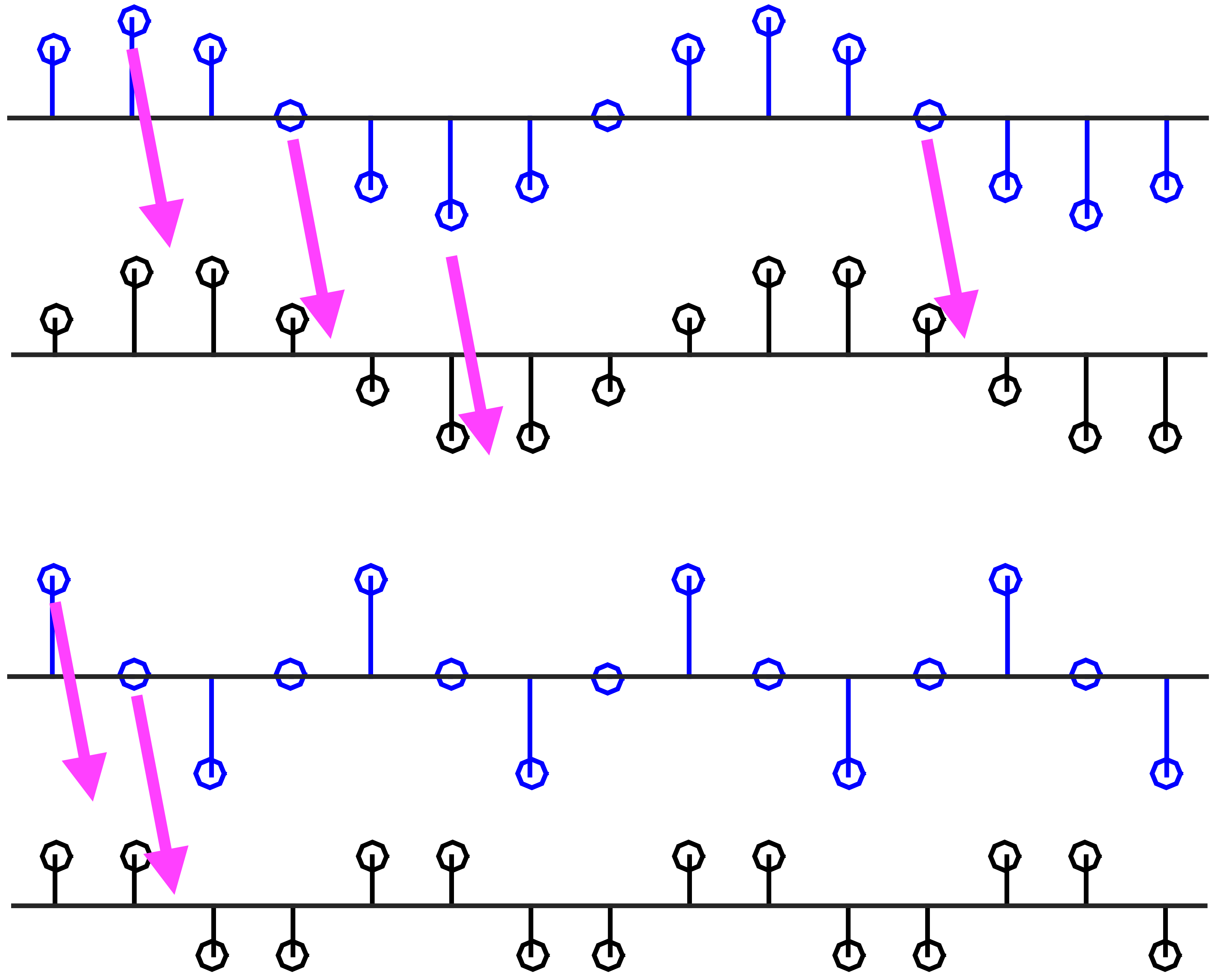
$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$



$$\tau_D: \quad \frac{\Delta t}{2} (?)$$

Group Delay Examples

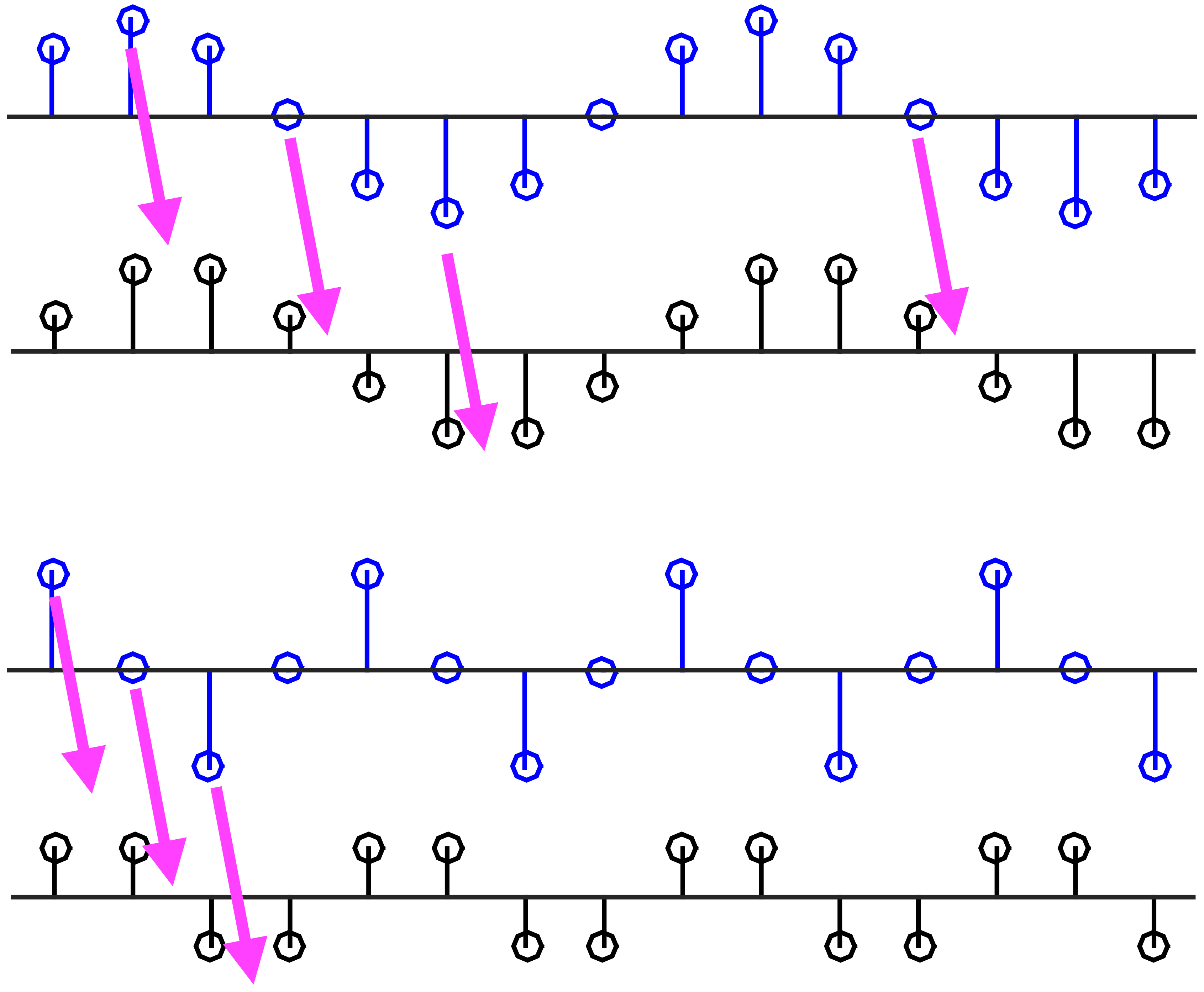
$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$



τ_D: $\frac{\Delta t}{2}$ (?)

Group Delay Examples

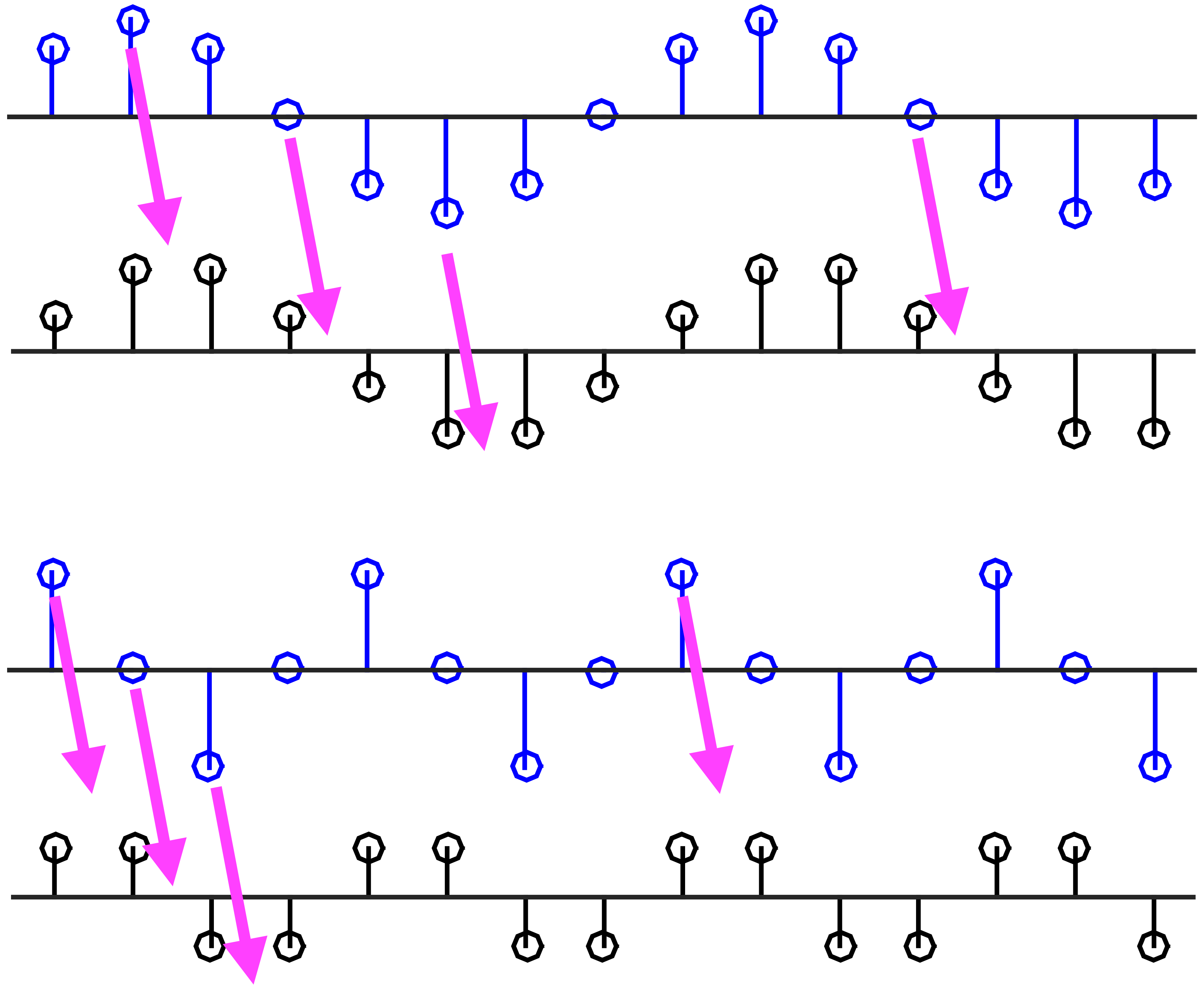
$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t-\Delta t]$$



$$\tau_D: \quad \frac{\Delta t}{2}(?)$$

Group Delay Examples

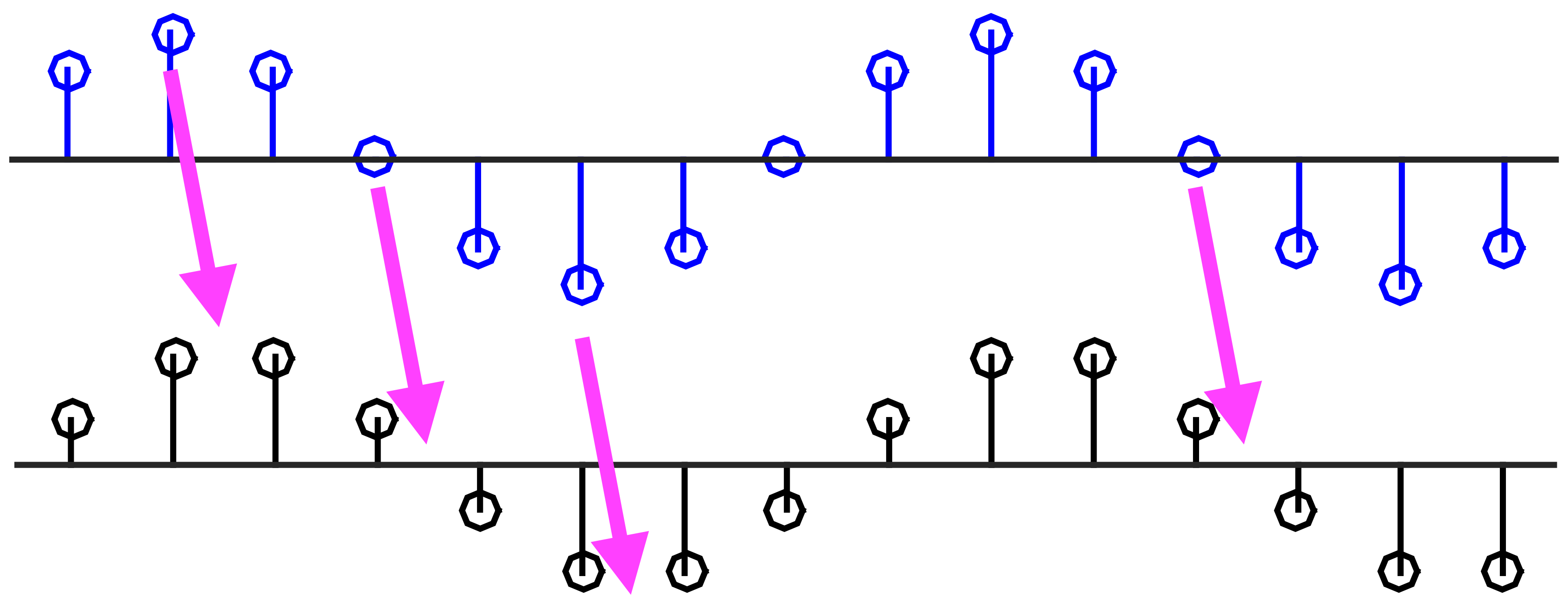
$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t - \Delta t]$$



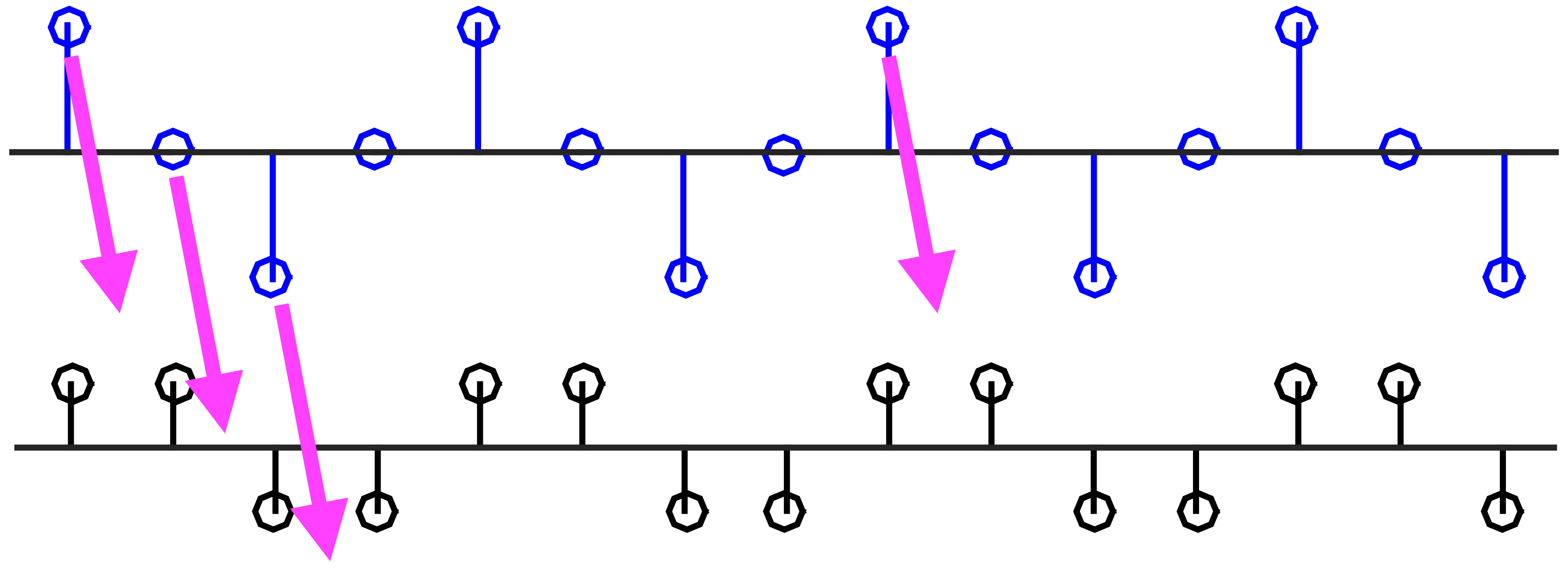
τ_D: $\frac{\Delta t}{2}$ (?)

Group Delay Examples

$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t - \Delta t]$$



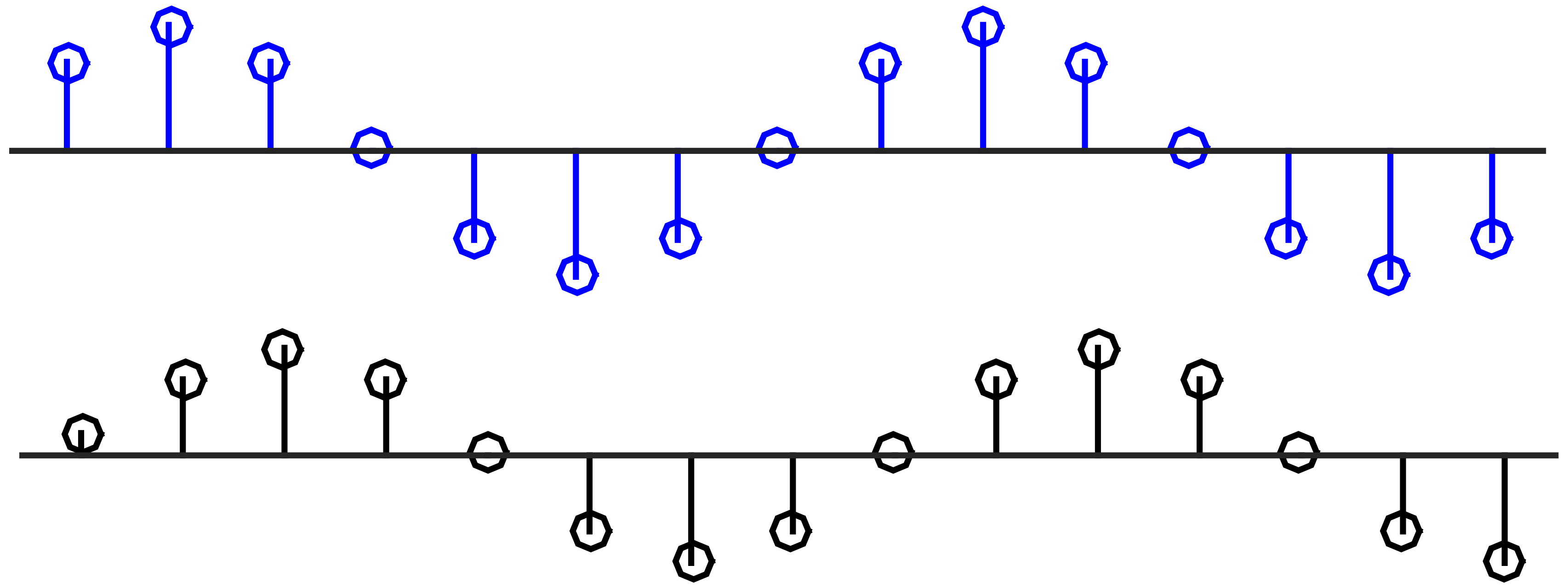
τ_D: $\frac{\Delta t}{2}$(?)



τ_D: $\frac{\Delta t}{2}$(?)

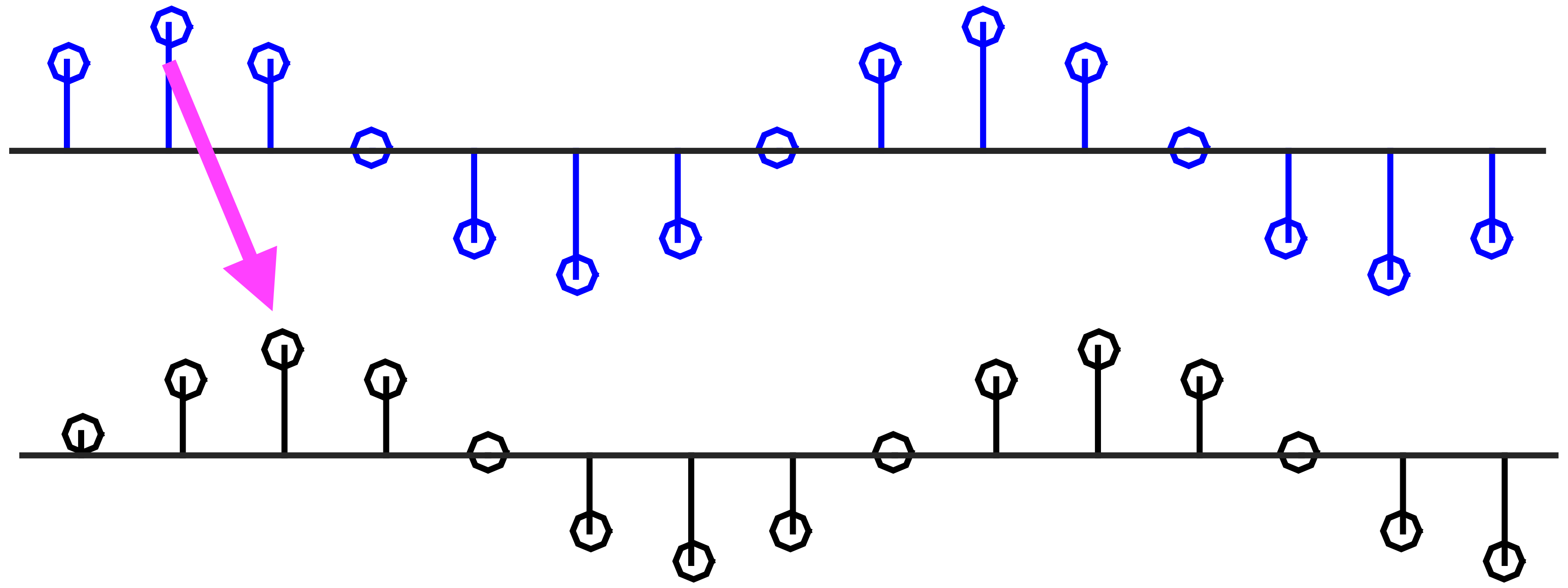
Group Delay Examples

$$y[t] = \frac{1}{4}x[t] + \frac{1}{2}x[t-\Delta t] + \frac{1}{4}x[t-2\Delta t]$$



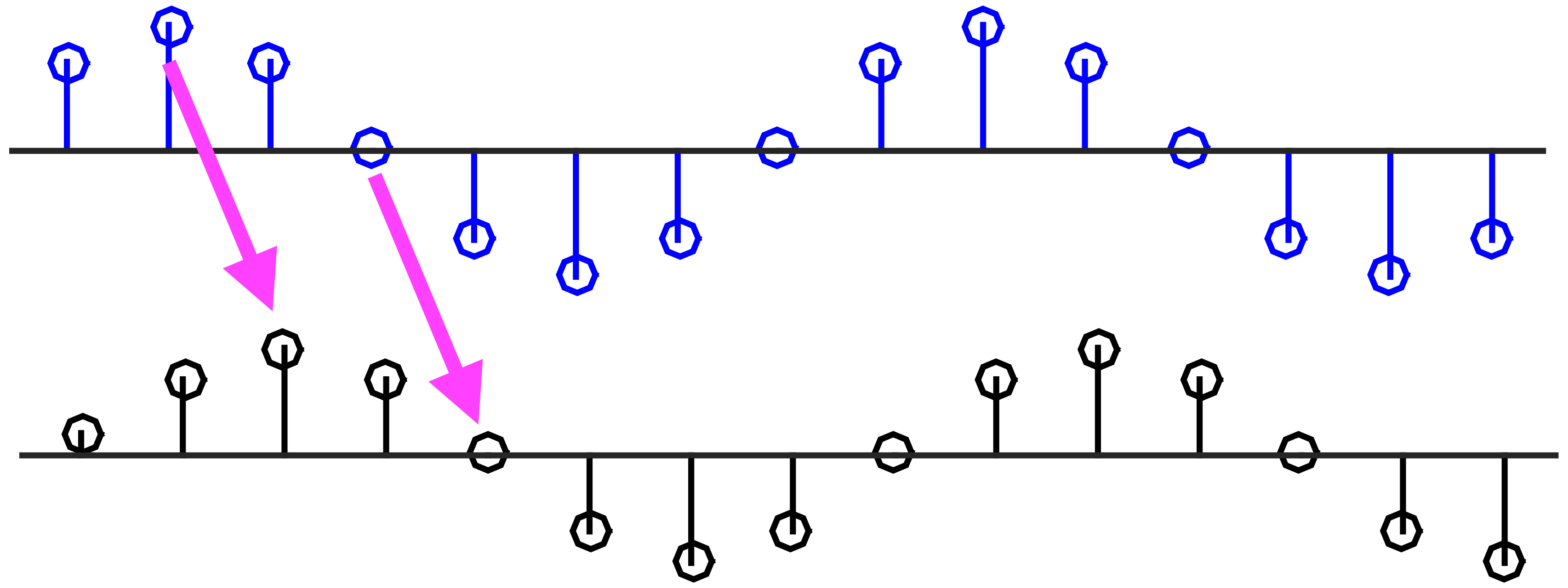
Group Delay Examples

$$y[t] = \frac{1}{4}x[t] + \frac{1}{2}x[t-\Delta t] + \frac{1}{4}x[t-2\Delta t]$$



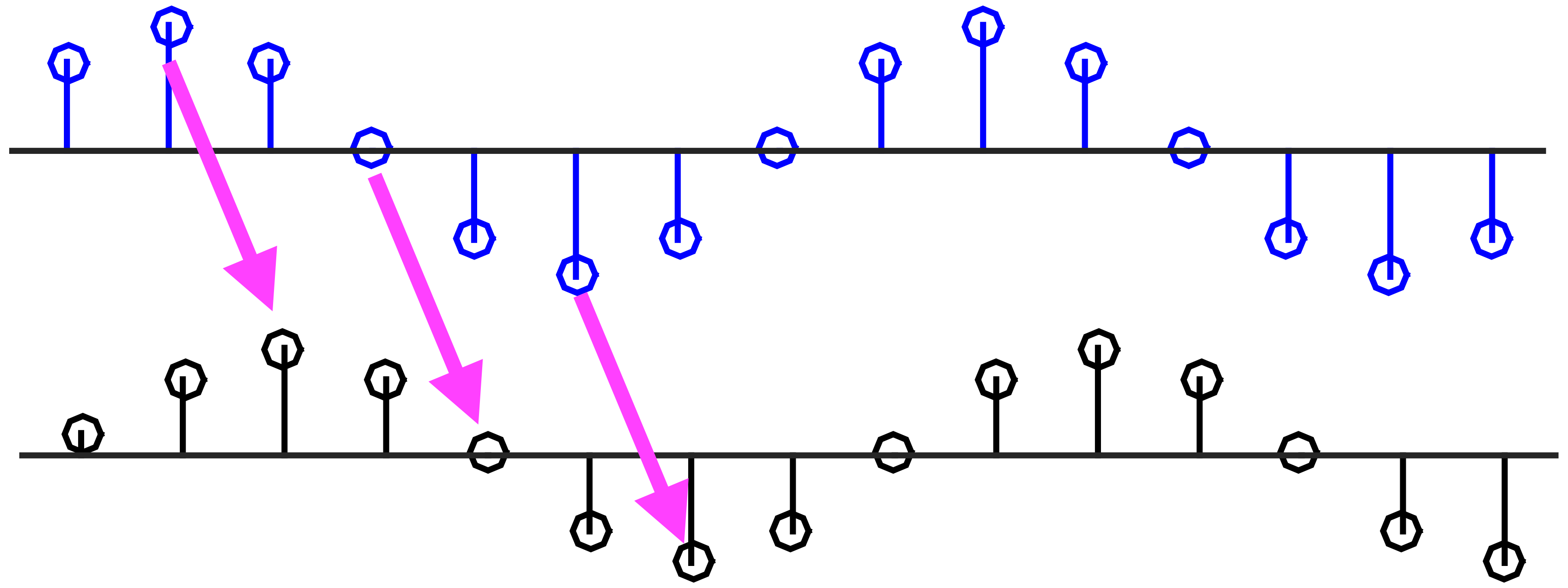
Group Delay Examples

$$y[t] = \frac{1}{4}x[t] + \frac{1}{2}x[t-\Delta t] + \frac{1}{4}x[t-2\Delta t]$$



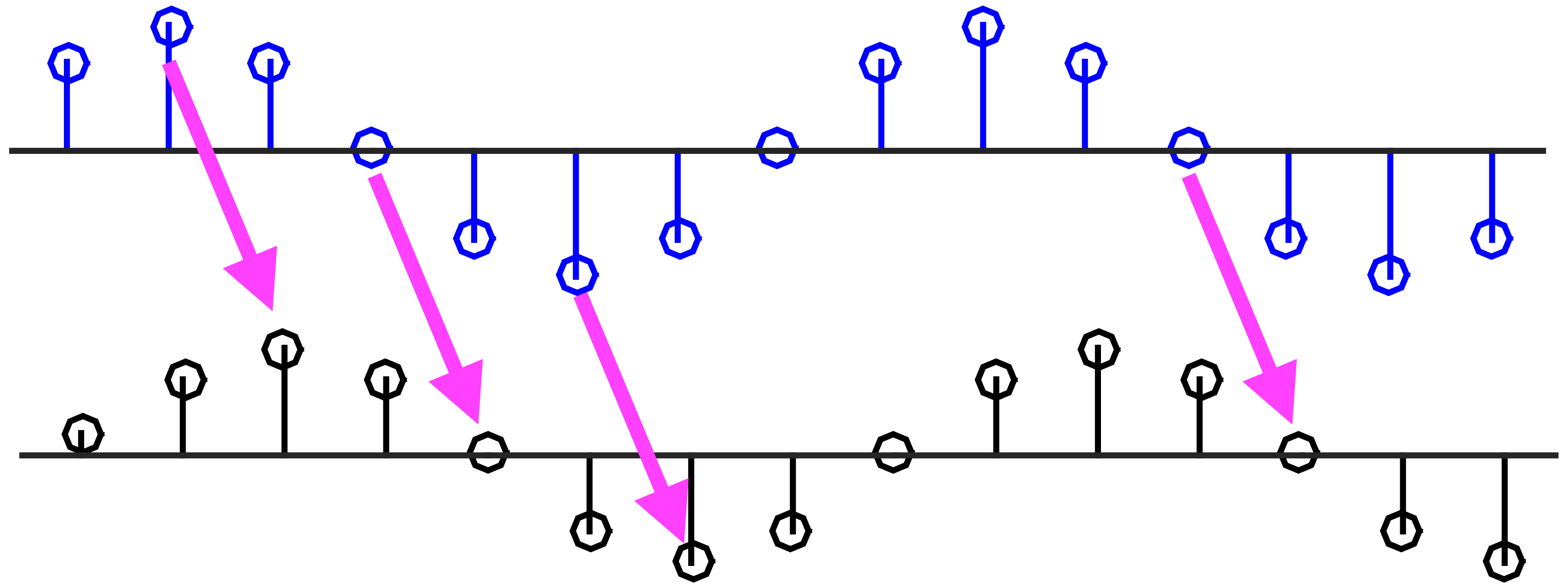
Group Delay Examples

$$y[t] = \frac{1}{4}x[t] + \frac{1}{2}x[t-\Delta t] + \frac{1}{4}x[t-2\Delta t]$$



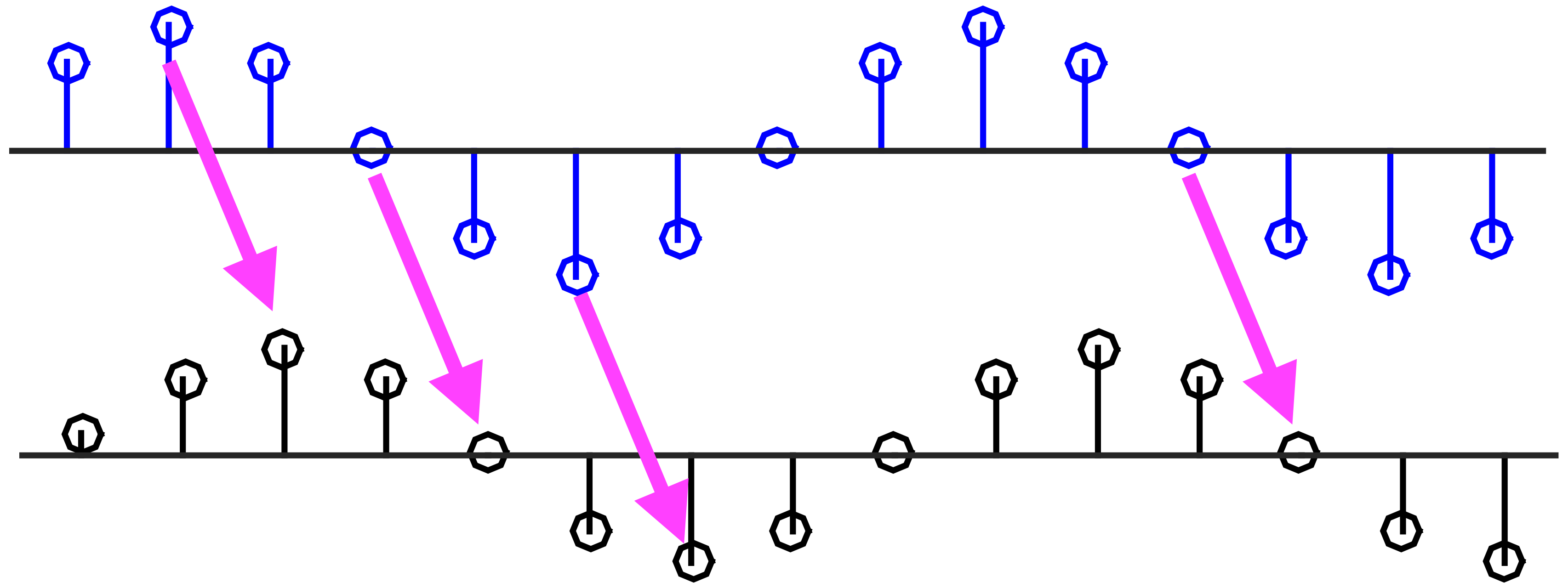
Group Delay Examples

$$y[t] = \frac{1}{4}x[t] + \frac{1}{2}x[t-\Delta t] + \frac{1}{4}x[t-2\Delta t]$$



Group Delay Examples

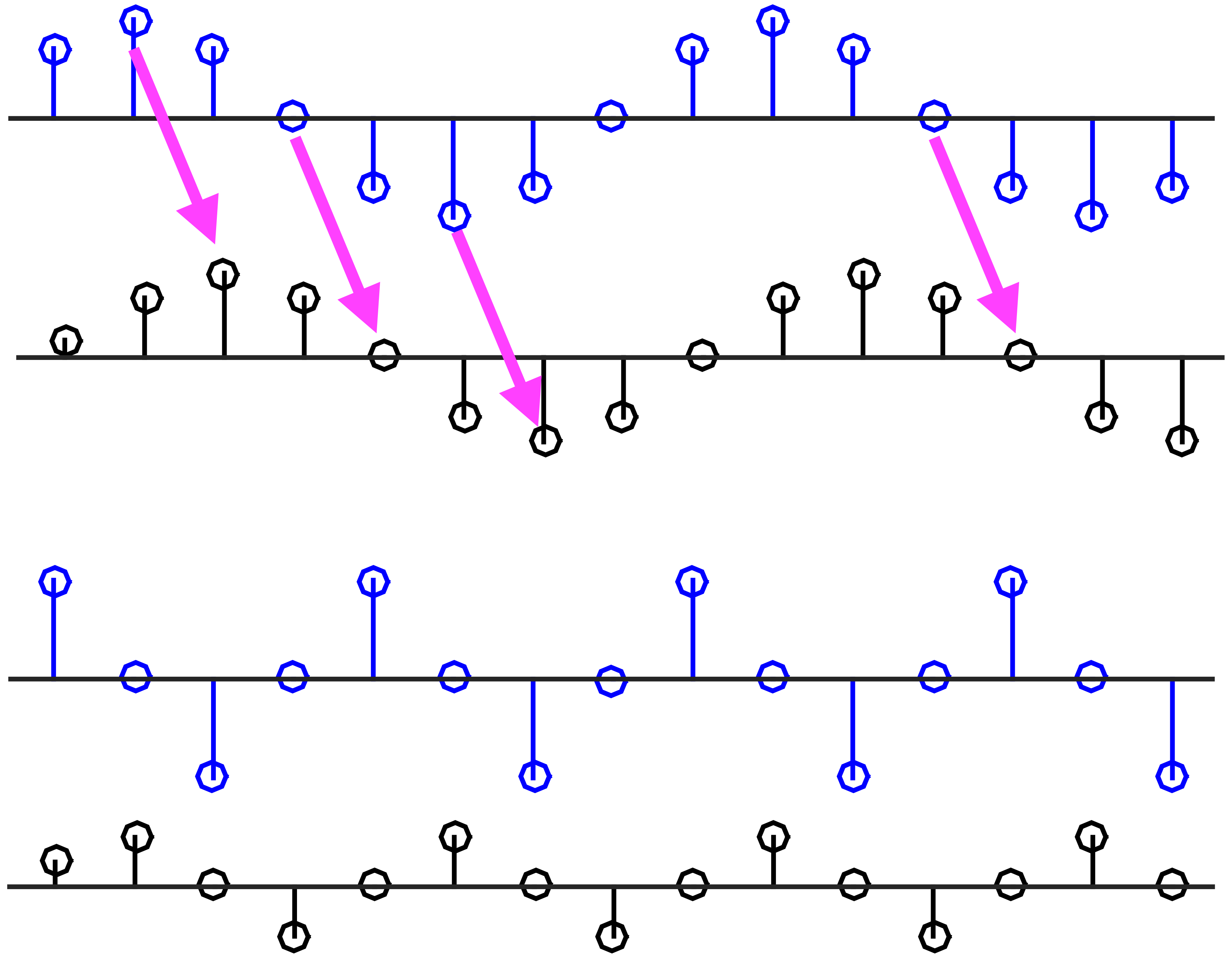
$$y[t] = \frac{1}{4}x[t] + \frac{1}{2}x[t-\Delta t] + \frac{1}{4}x[t-2\Delta t]$$



τ_D: Δt

Group Delay Examples

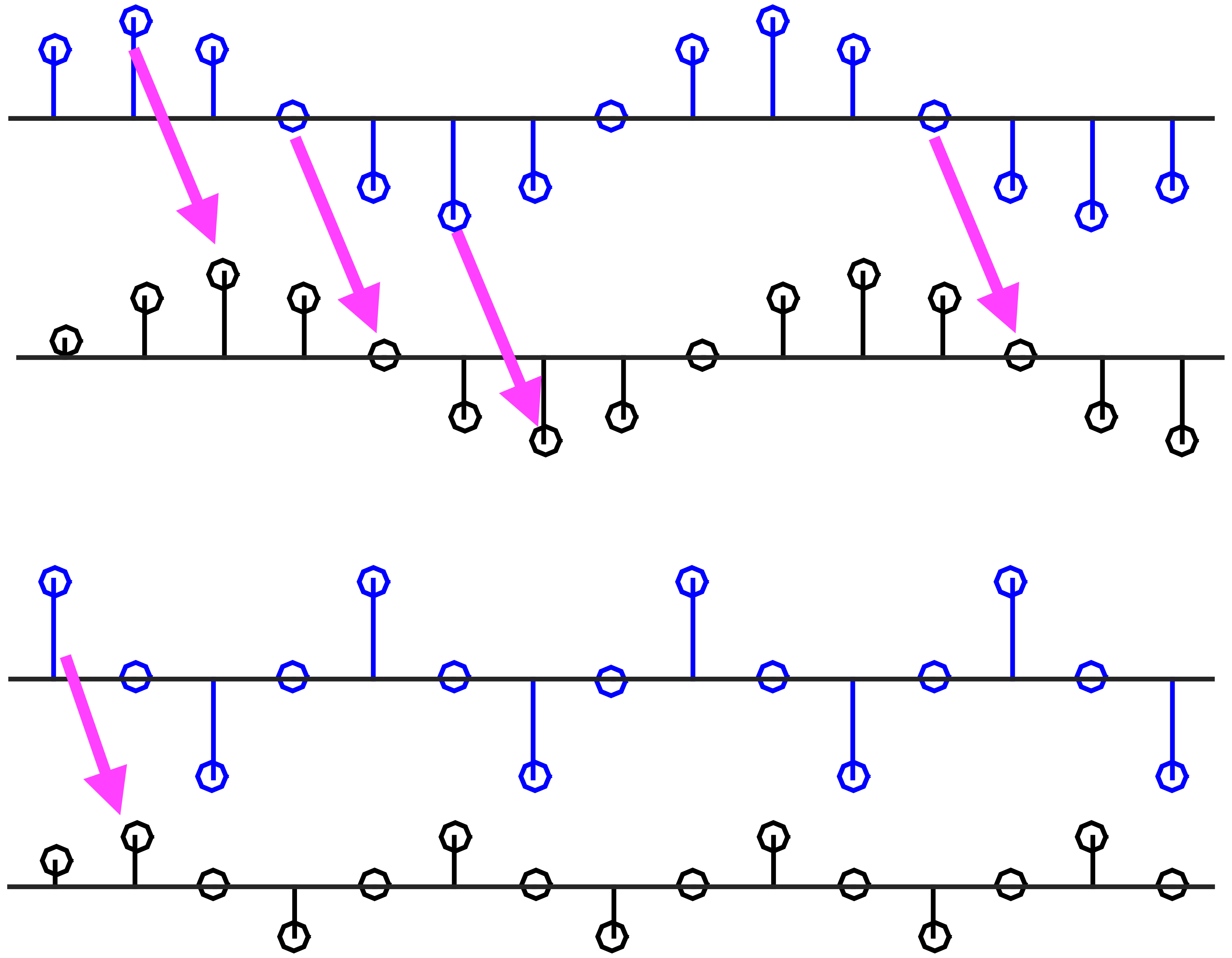
$$y[t]=\frac{1}{4}x[t]+\frac{1}{2}x[t-\Delta t]+\frac{1}{4}x[t-2\Delta t]$$



τ_D: Δt

Group Delay Examples

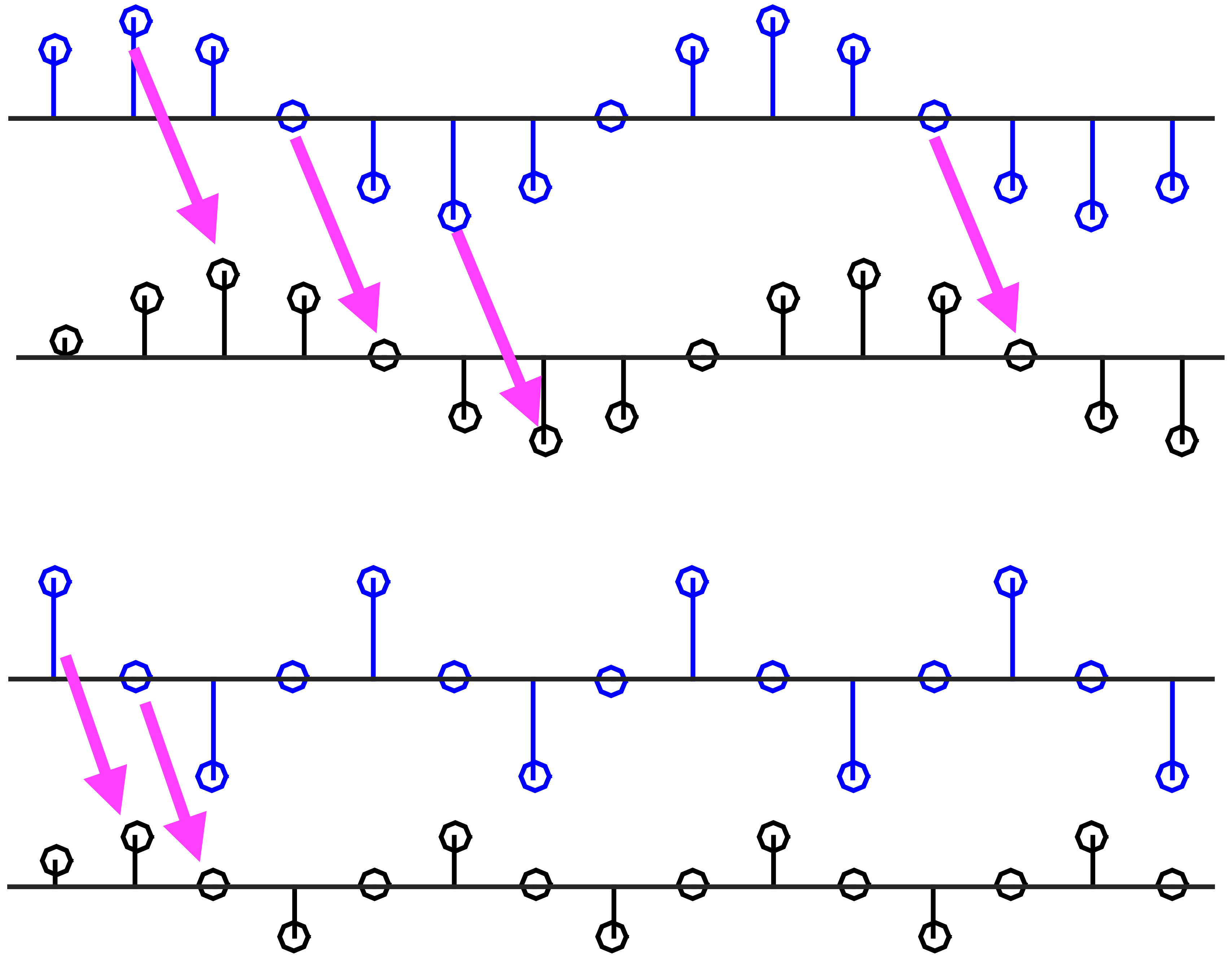
$$y[t] = \frac{1}{4}x[t] + \frac{1}{2}x[t-\Delta t] + \frac{1}{4}x[t-2\Delta t]$$



τ_D: Δt

Group Delay Examples

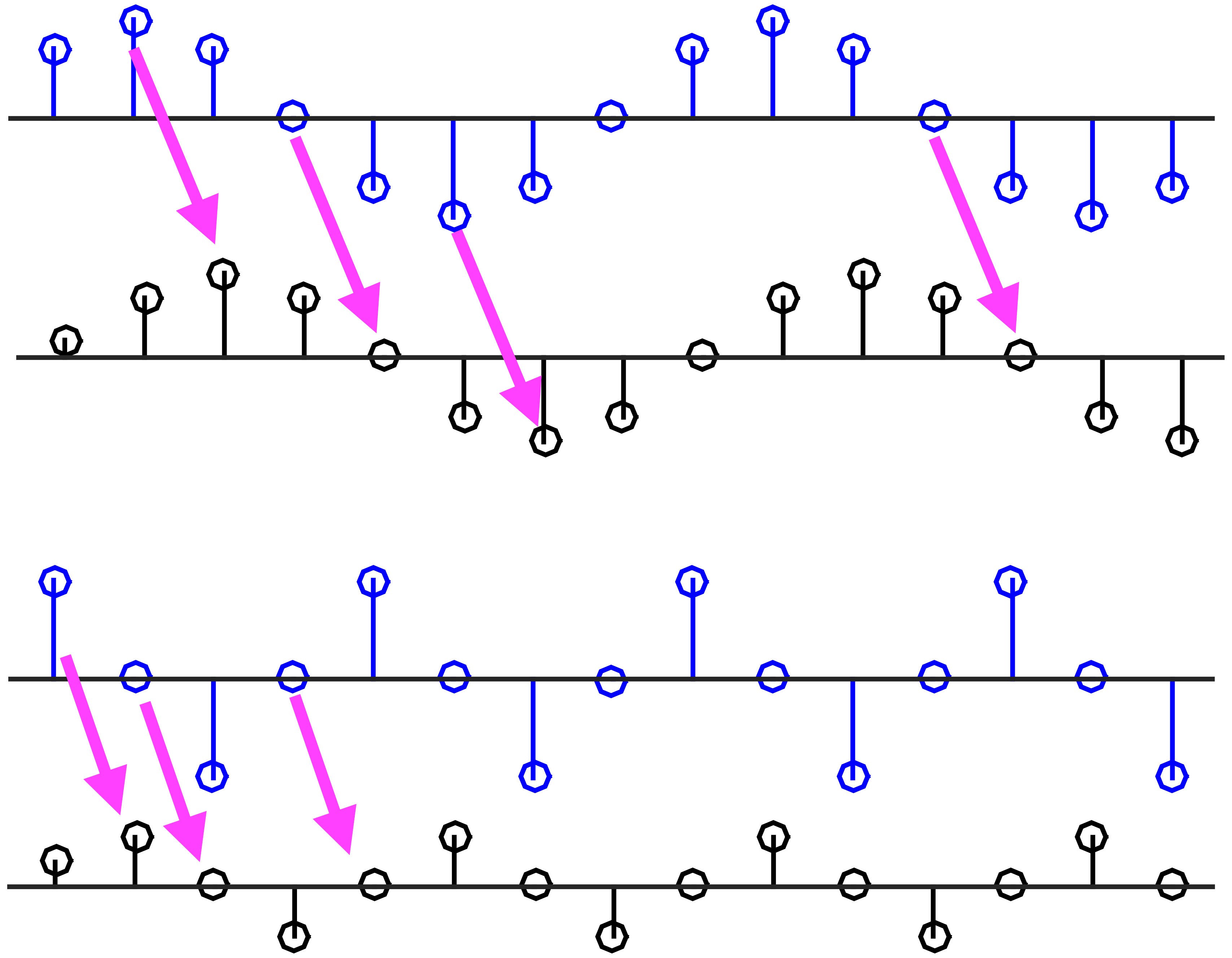
$$y[t] = \frac{1}{4}x[t] + \frac{1}{2}x[t-\Delta t] + \frac{1}{4}x[t-2\Delta t]$$



τ_D: Δt

Group Delay Examples

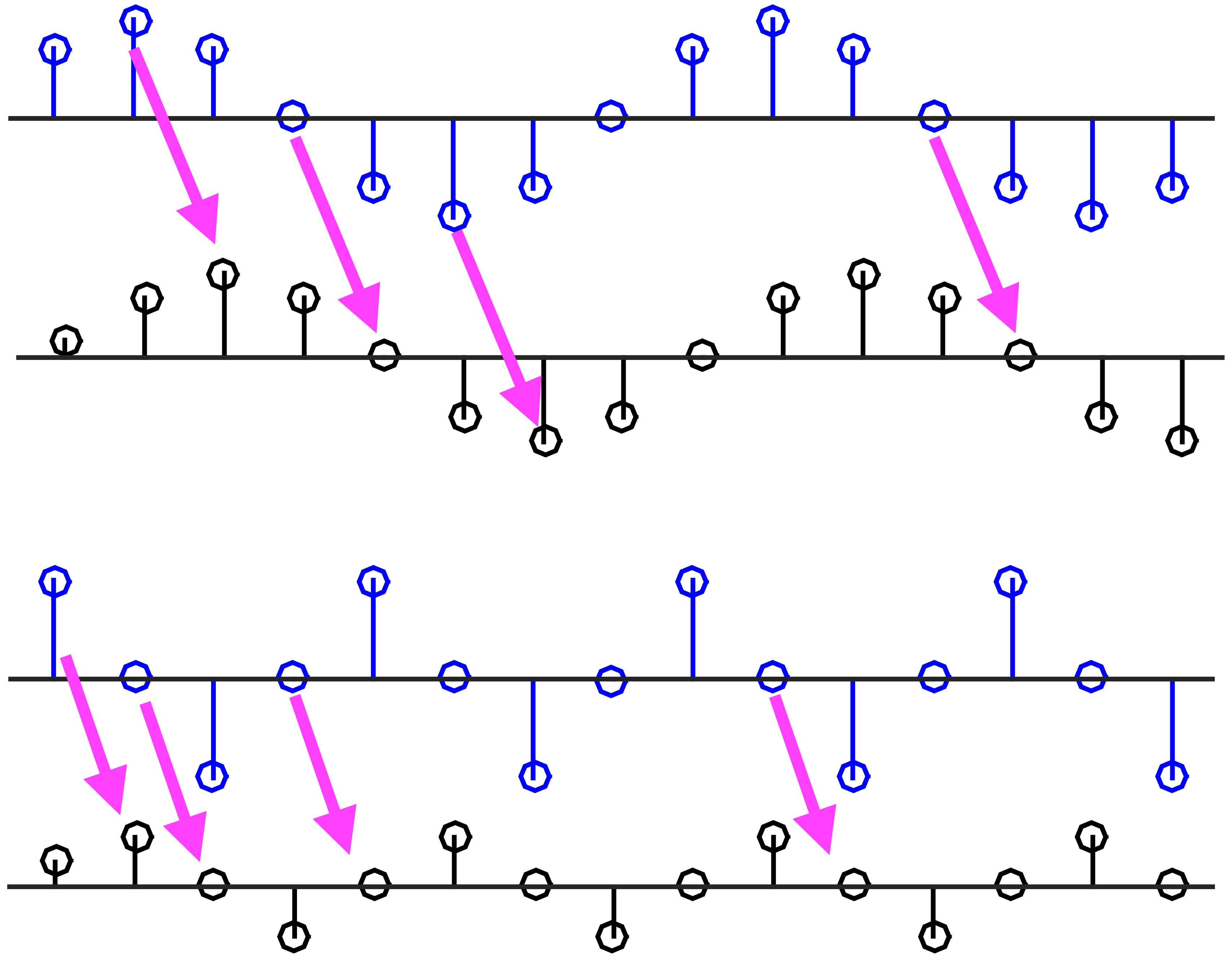
$$y[t] = \frac{1}{4}x[t] + \frac{1}{2}x[t-\Delta t] + \frac{1}{4}x[t-2\Delta t]$$



τ_D: Δt

Group Delay Examples

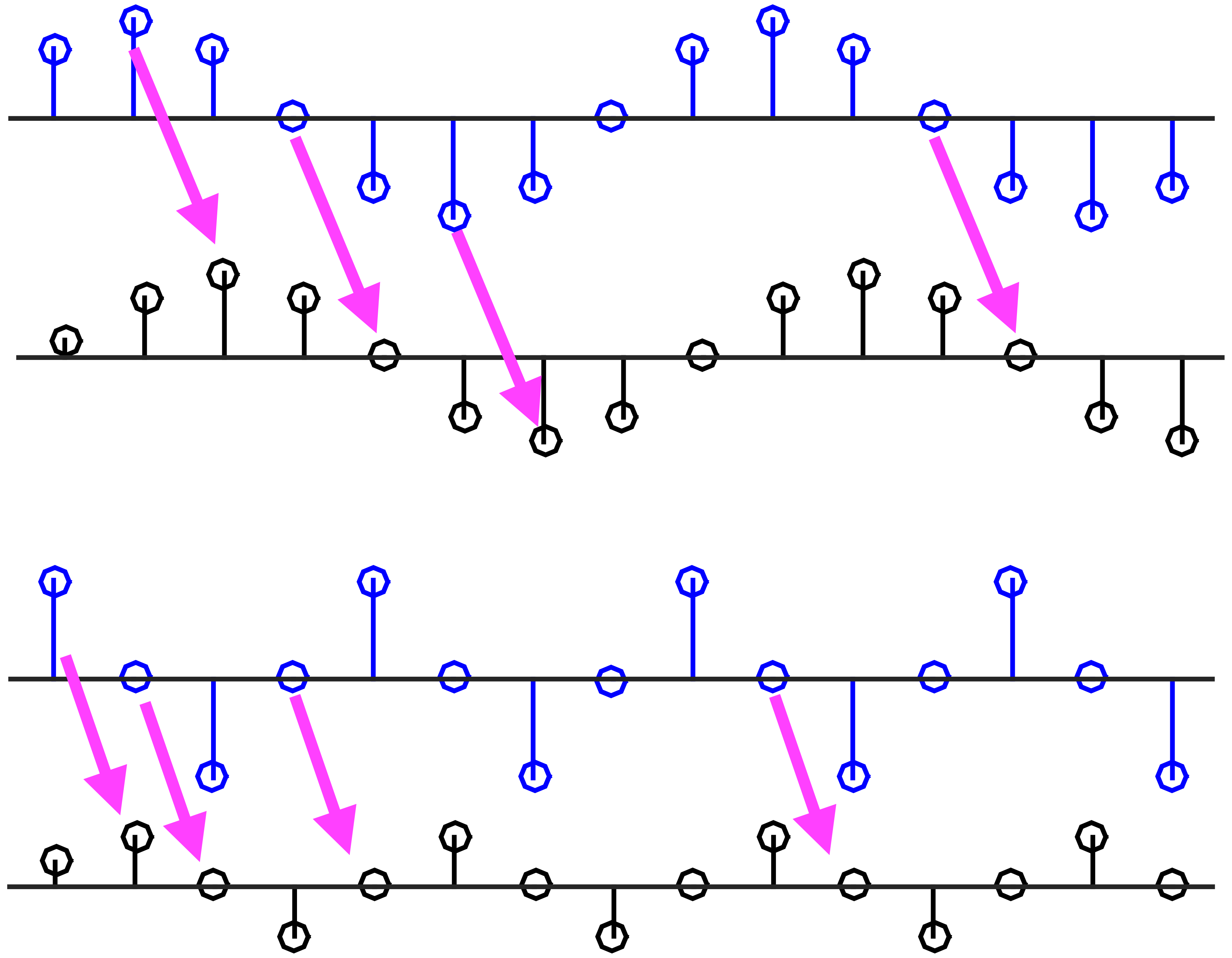
$$y[t] = \frac{1}{4}x[t] + \frac{1}{2}x[t-\Delta t] + \frac{1}{4}x[t-2\Delta t]$$



τ_D: Δt

Group Delay Examples

$$y[t] = \frac{1}{4}x[t] + \frac{1}{2}x[t-\Delta t] + \frac{1}{4}x[t-2\Delta t]$$



τ_D: Δt

τ_D: Δt

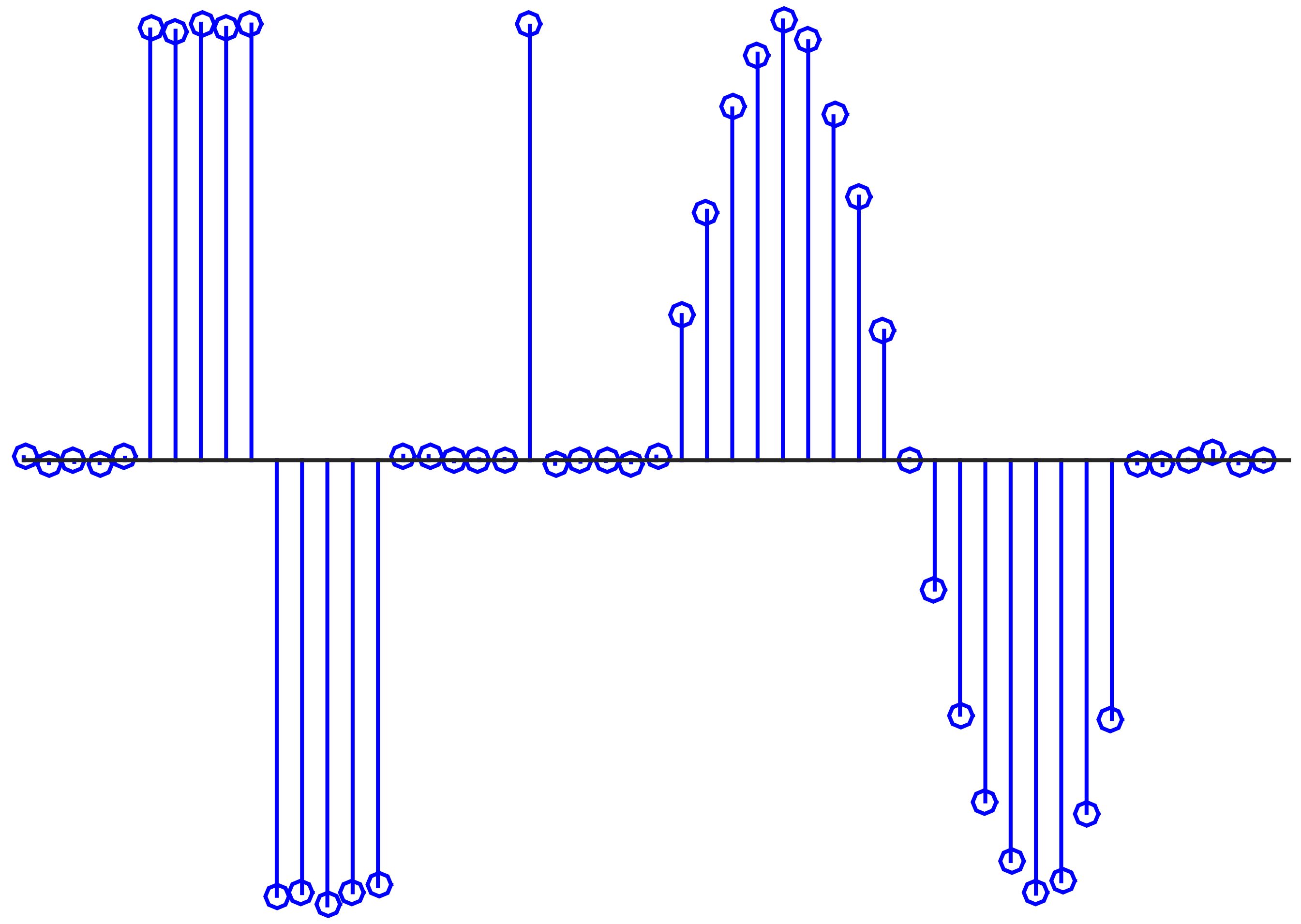
Group Delay: FIR filters

- Group delay corresponds to “average” delay imparted by *time-shifted* filter terms.
- The group delay of an FIR filter does not depend on frequency.
- The *order* of an FIR filter, N_{order}, is the number of time shifts of the *most delayed* component (same as the length of the filter, minus 1).
- The group delay of an FIR filter is $\Delta t \times N_{order}/2$.
  - The higher the order, the longer the group delay
  - Calculating latencies? You may need to compensate (especially for *peak* latencies).
  - Smaller Δt = smaller delay. So if possible, filter at high sampling frequency.

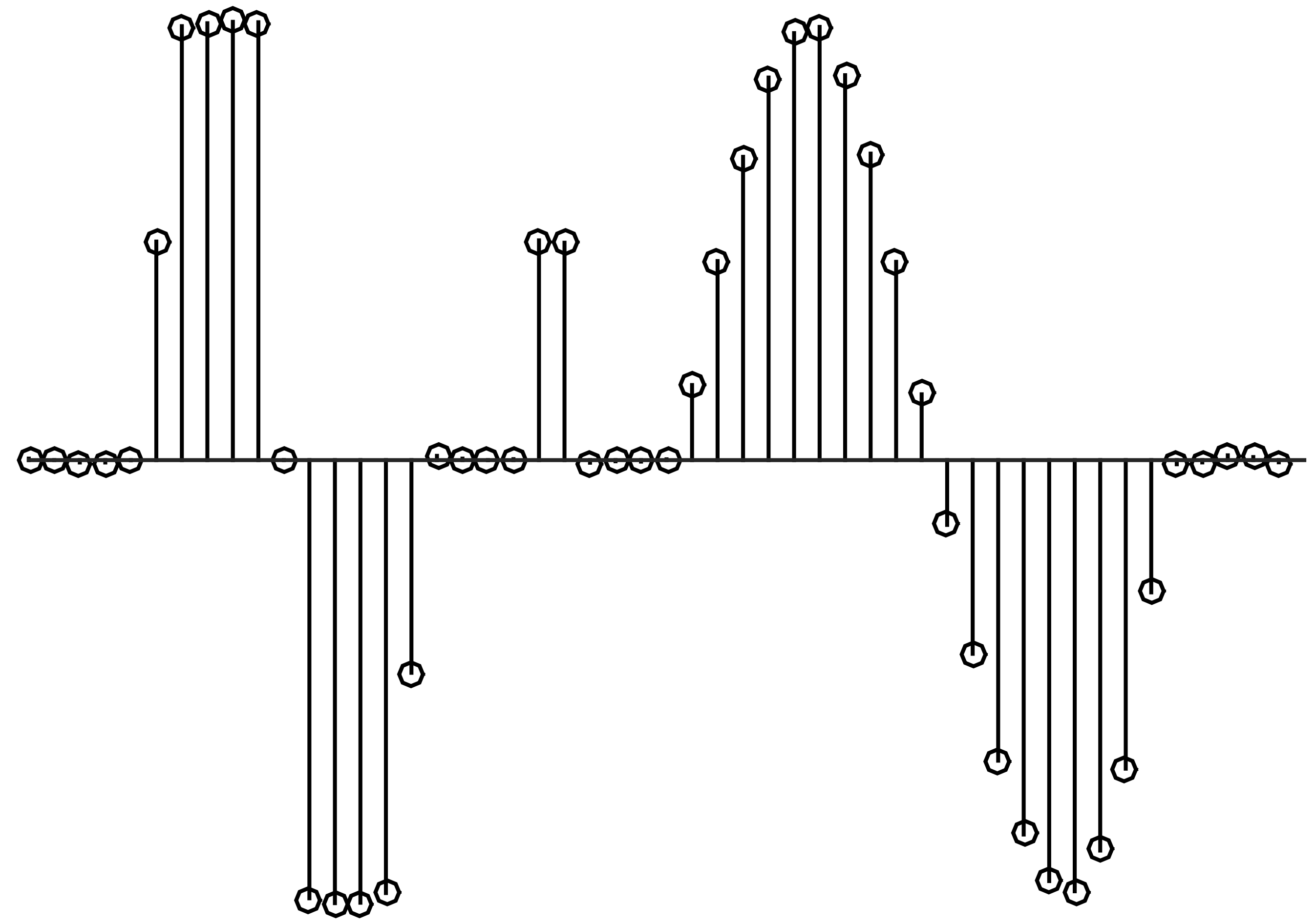
FIR Group Delay: General Signals

For non-sinusoidal (multi-frequency) signals, group delay still applies, but *how* it manifests depends on the specific signal features.

$x[t]$



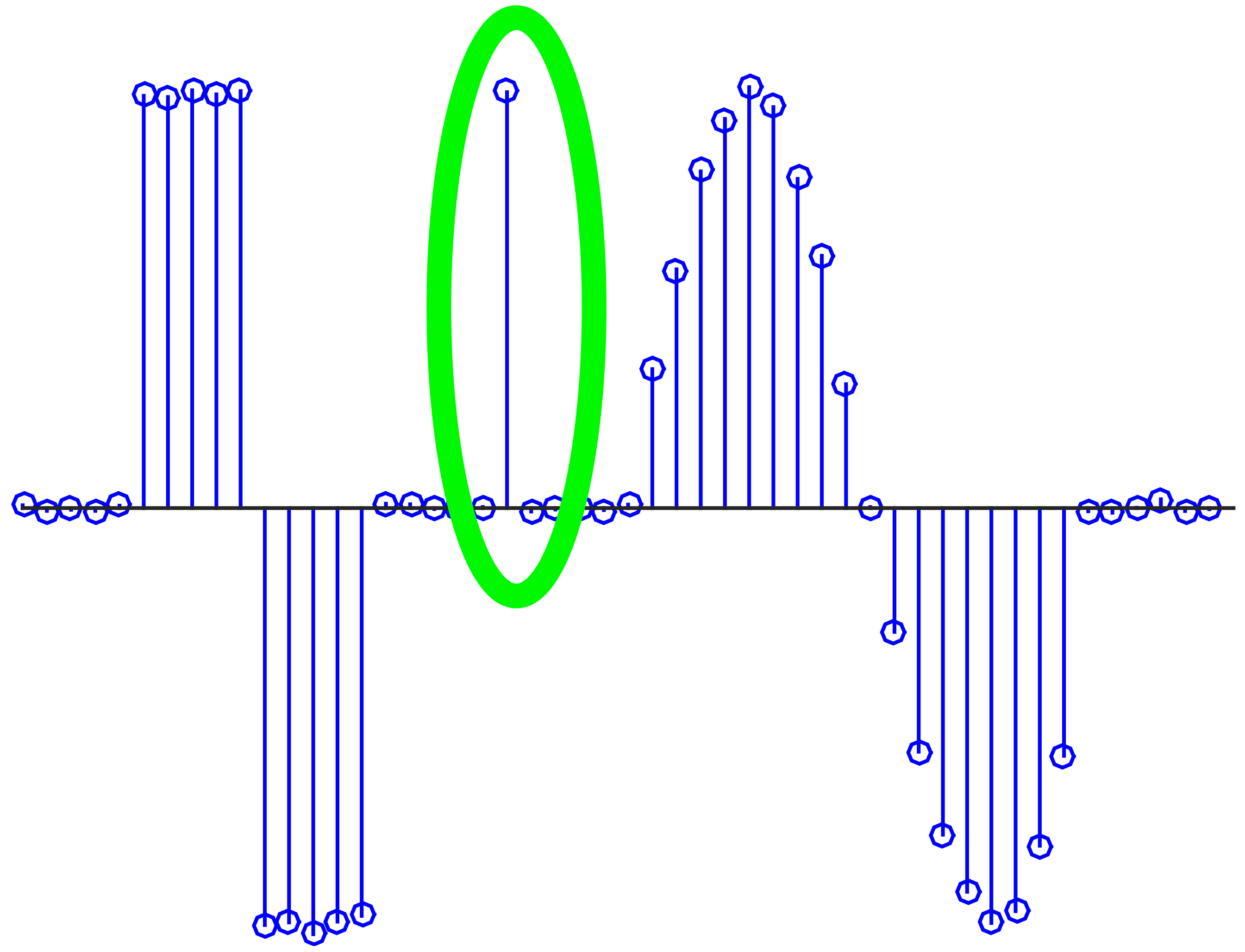
$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t - \Delta t]$$



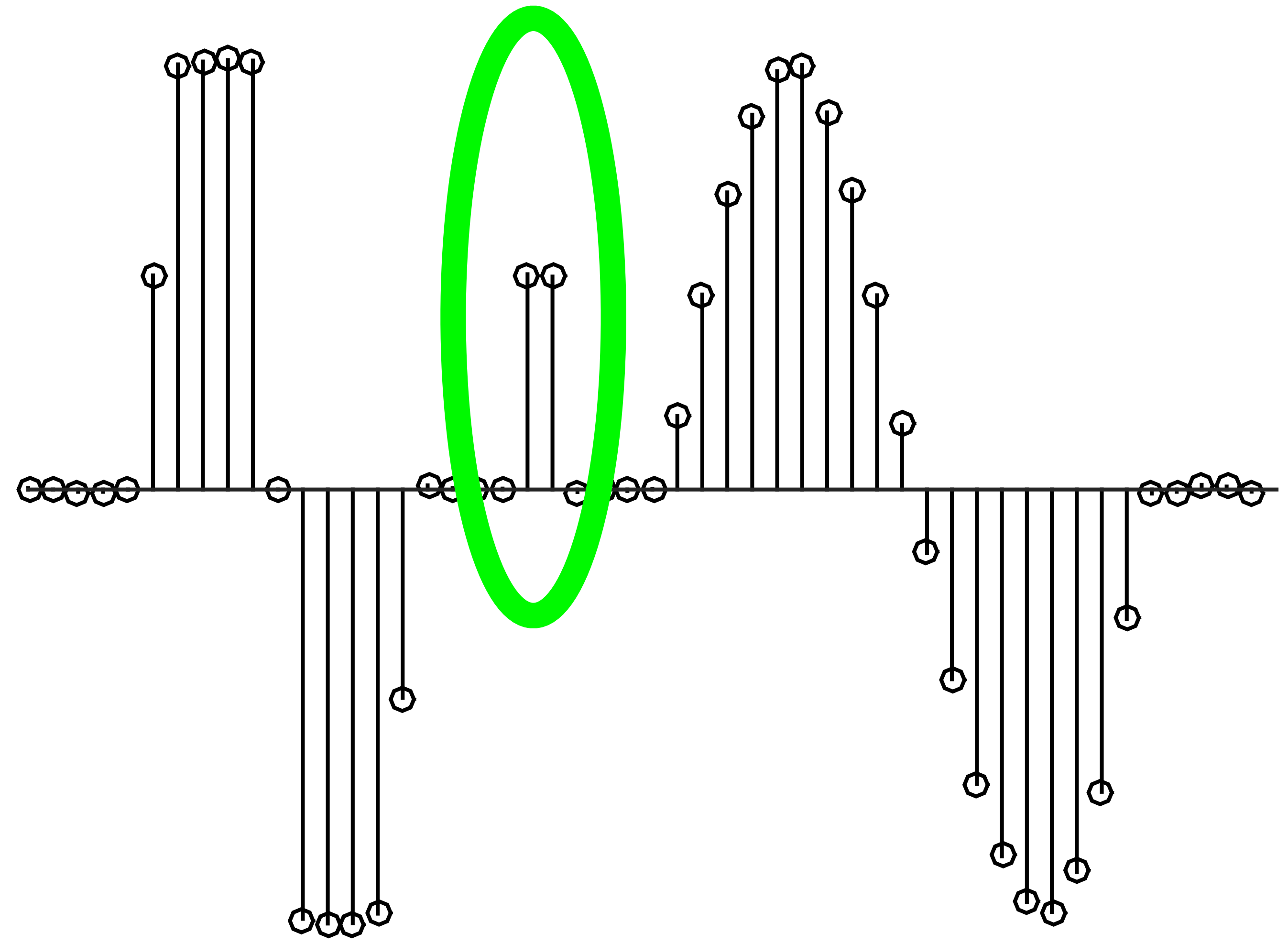
FIR Group Delay: General Signals

For non-sinusoidal (multi-frequency) signals, group delay still applies, but *how* it manifests depends on the specific signal features.

$x[t]$



$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t - \Delta t]$$

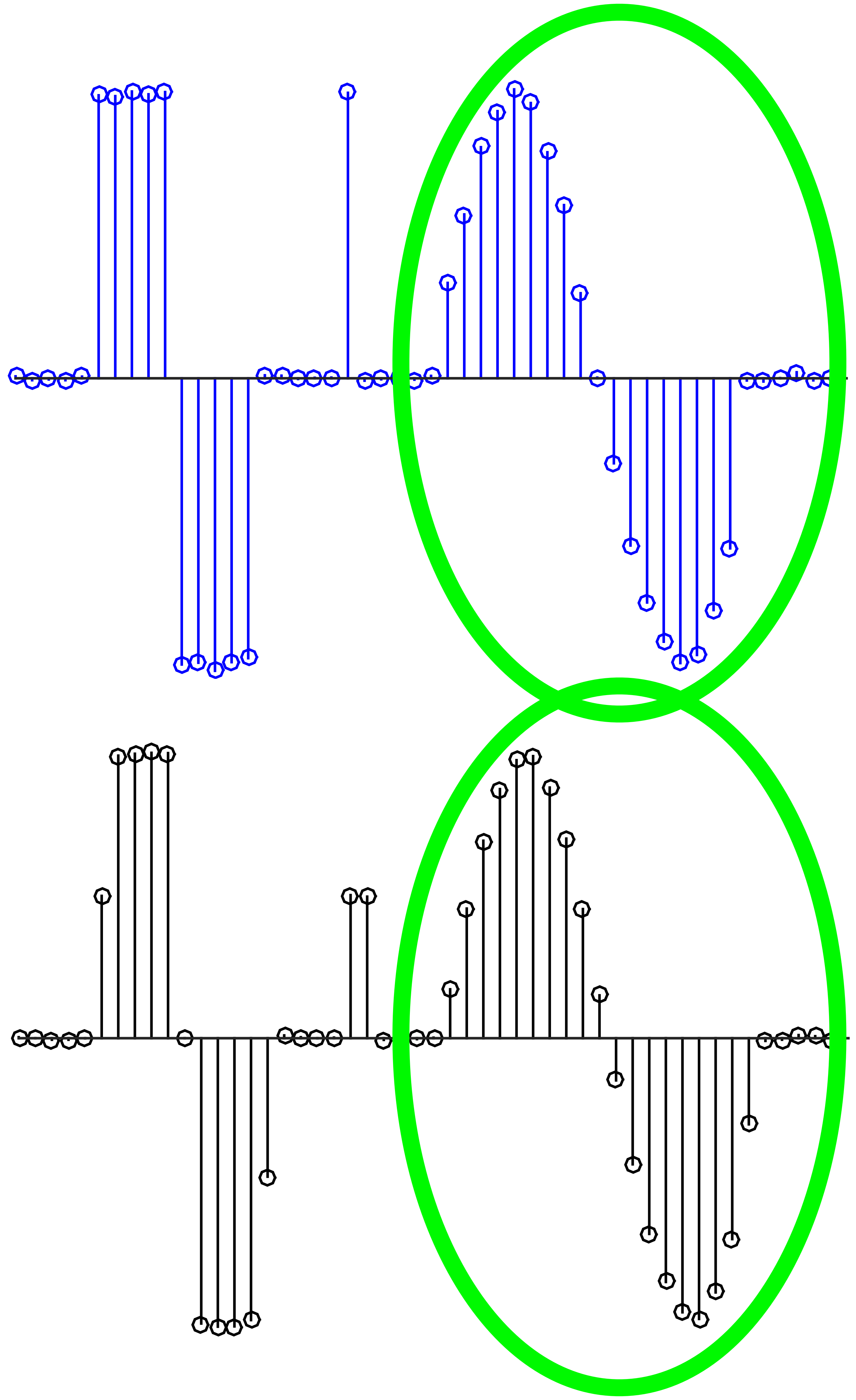


FIR Group Delay: General Signals

For non-sinusoidal (multi-frequency) signals, group delay still applies, but *how* it manifests depends on the specific signal features.

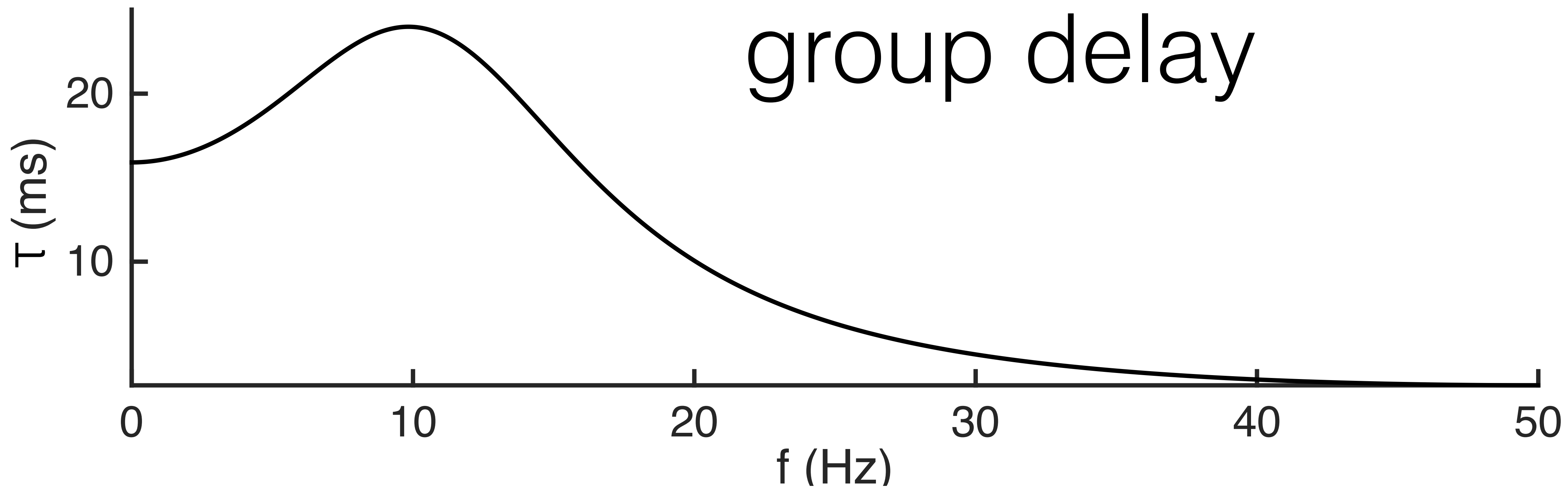
$$x[t]$$

$$y[t] = \frac{1}{2}x[t] + \frac{1}{2}x[t - \Delta t]$$



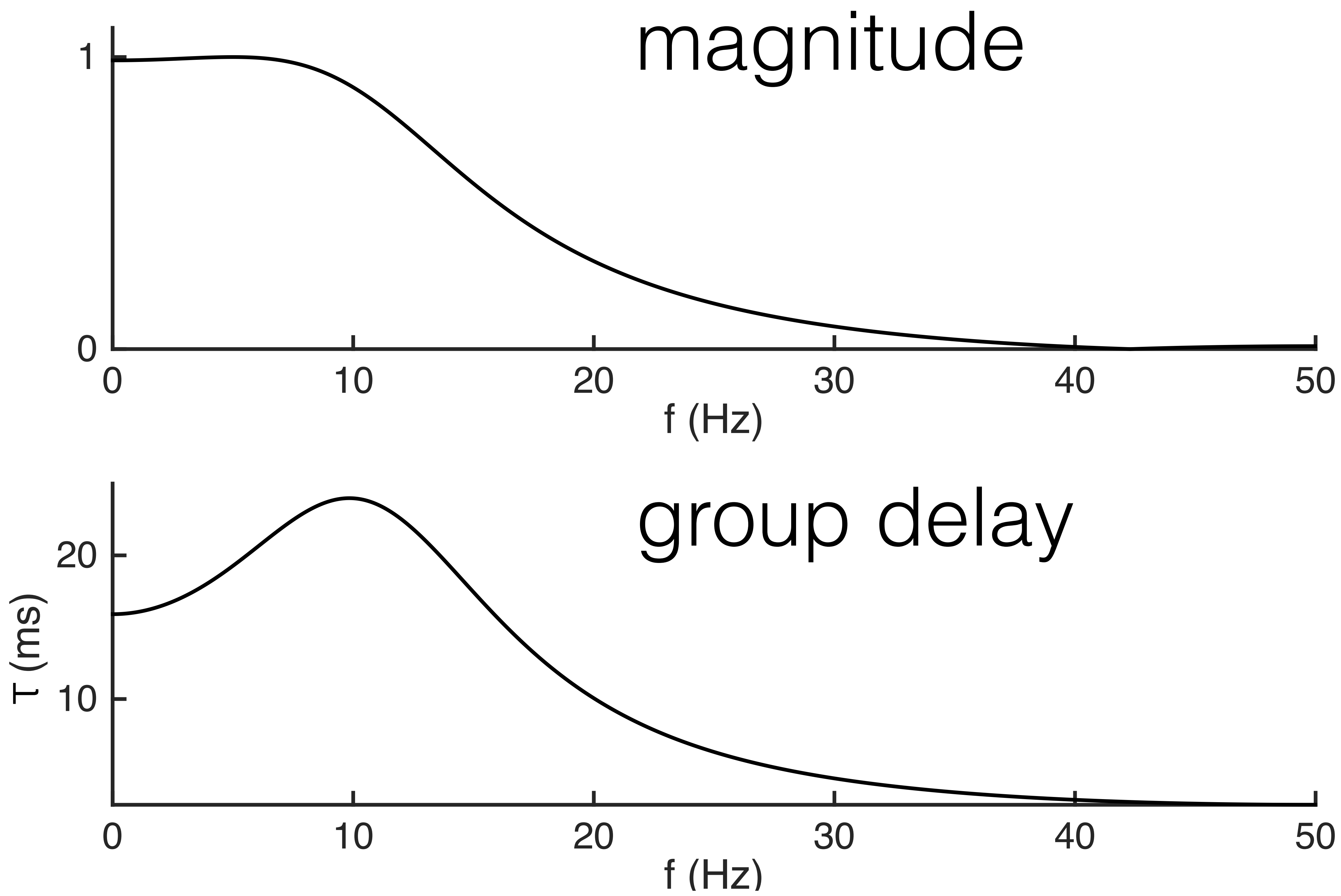
Group Delay: IIR filters

- The group delay of an IIR filter **does** depend on **frequency**.



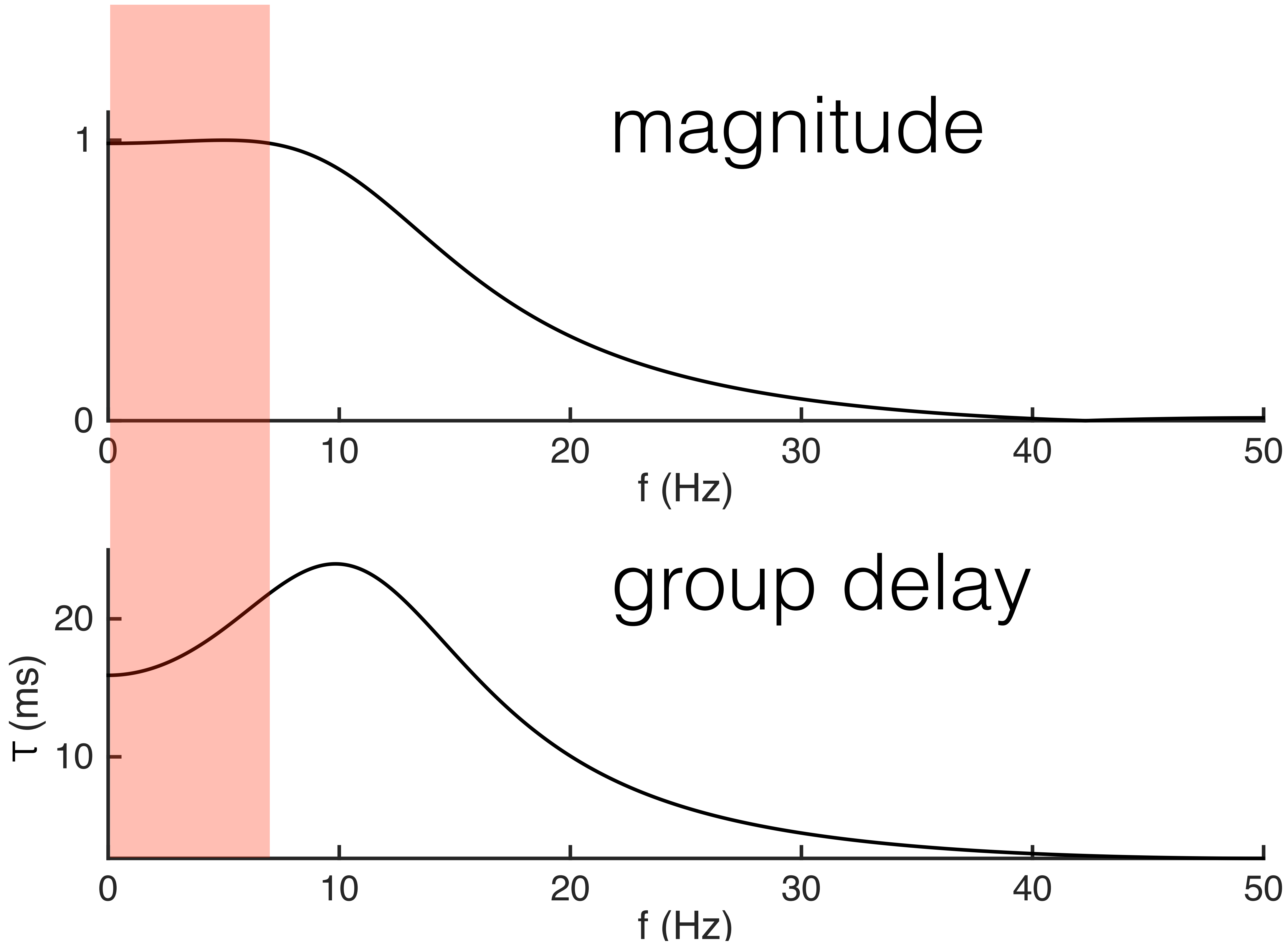
Group Delay: IIR filters

- The group delay of an IIR filter **does** depend on **frequency**.



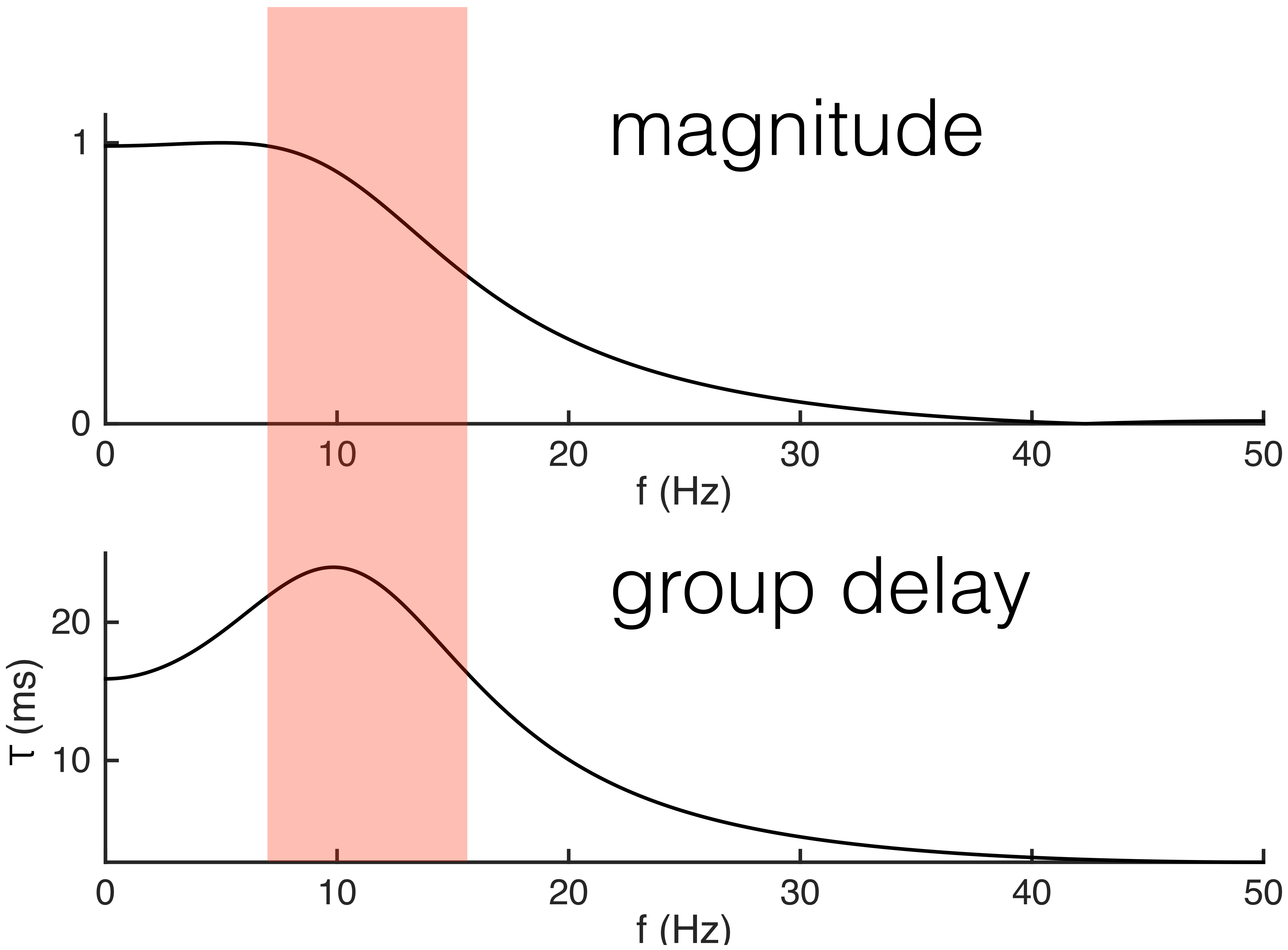
Group Delay: IIR filters

- The group delay of an IIR filter **does** depend on **frequency**.
- The group delay of an IIR filter is relatively constant over frequencies that are "passed".



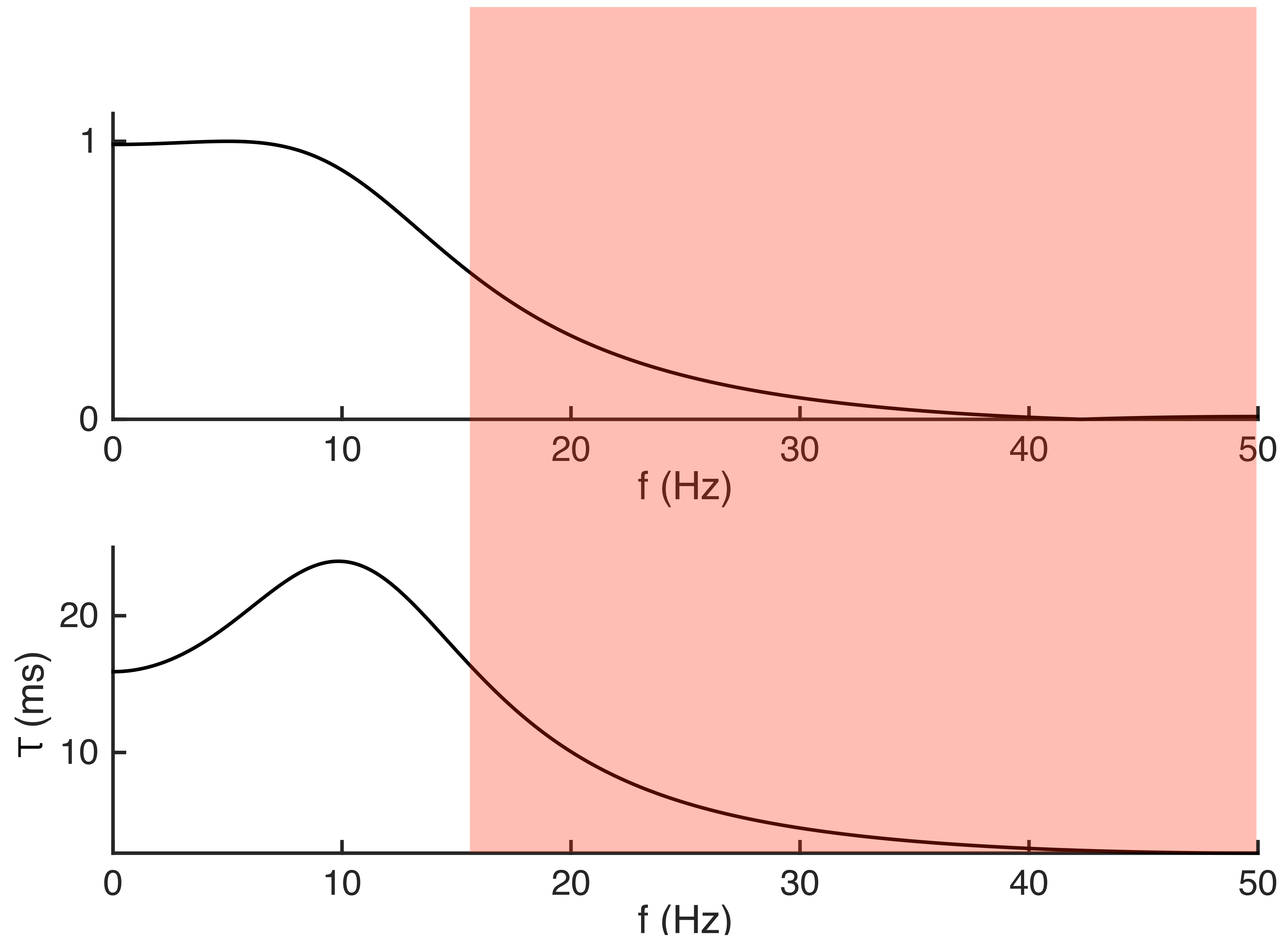
Group Delay: IIR filters

- The group delay of an IIR filter **does** depend on **frequency**.
- The group delay of an IIR filter is relatively constant over frequencies that are "passed".
- The group delay of an IIR filter is longest during the frequency transition.



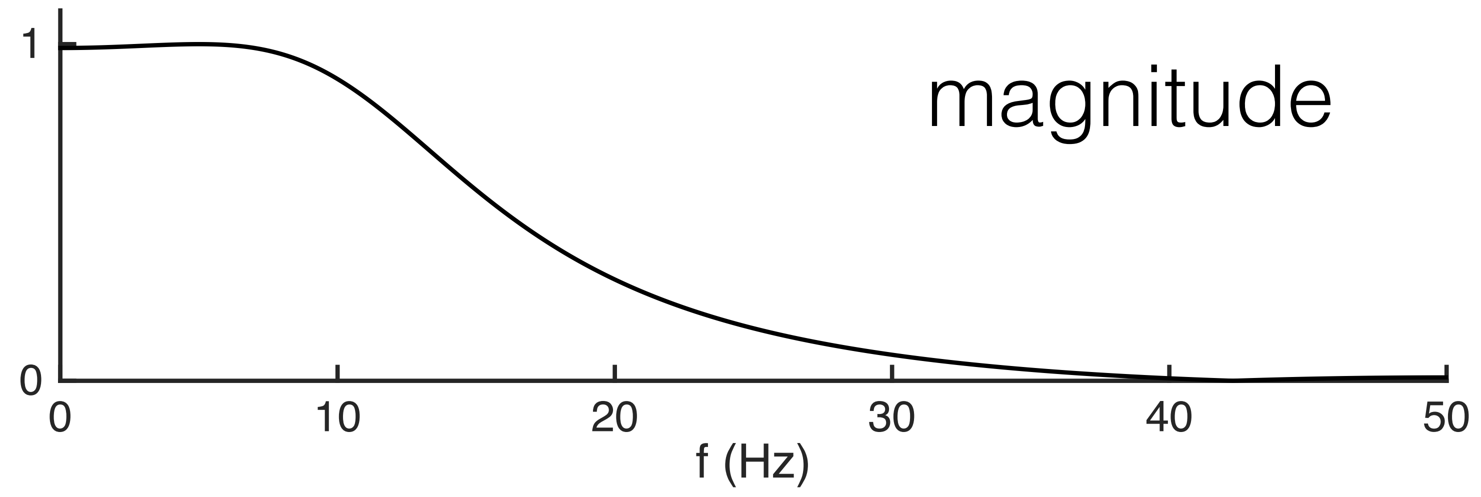
Group Delay: IIR filters

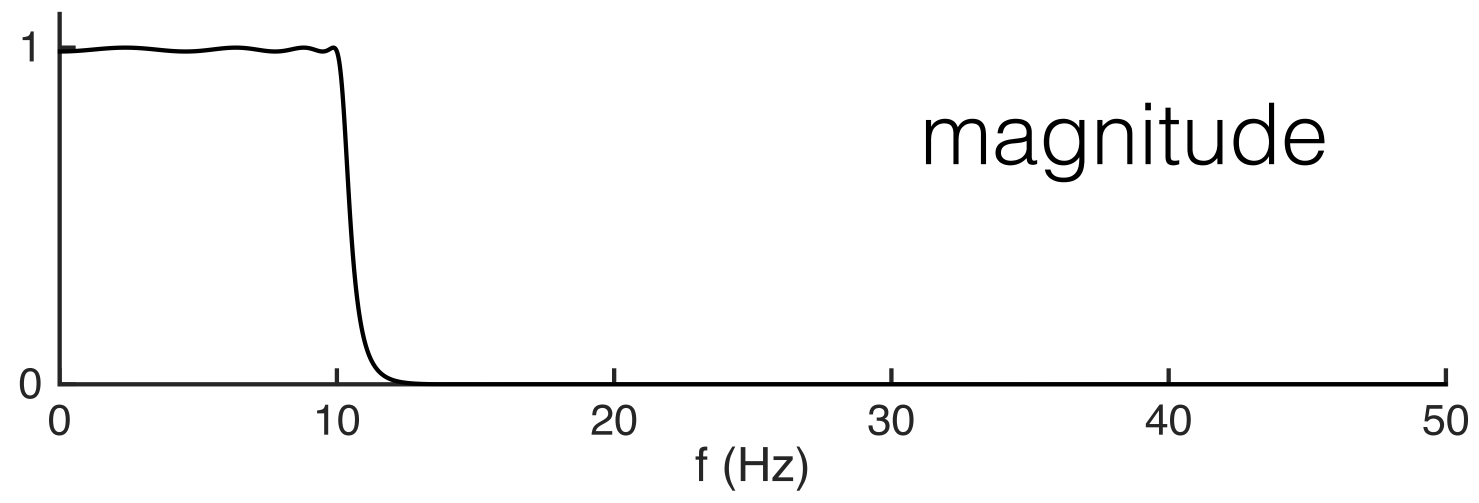
- The group delay of an IIR filter **does** depend on **frequency**.
- The group delay of an IIR filter is relatively constant over frequencies that are “passed”.
- The group delay of an IIR filter is longest during the frequency transition.
- The group delay of an IIR filter may be irrelevant over frequencies that are “stopped”.



Group Delay: IIR filters

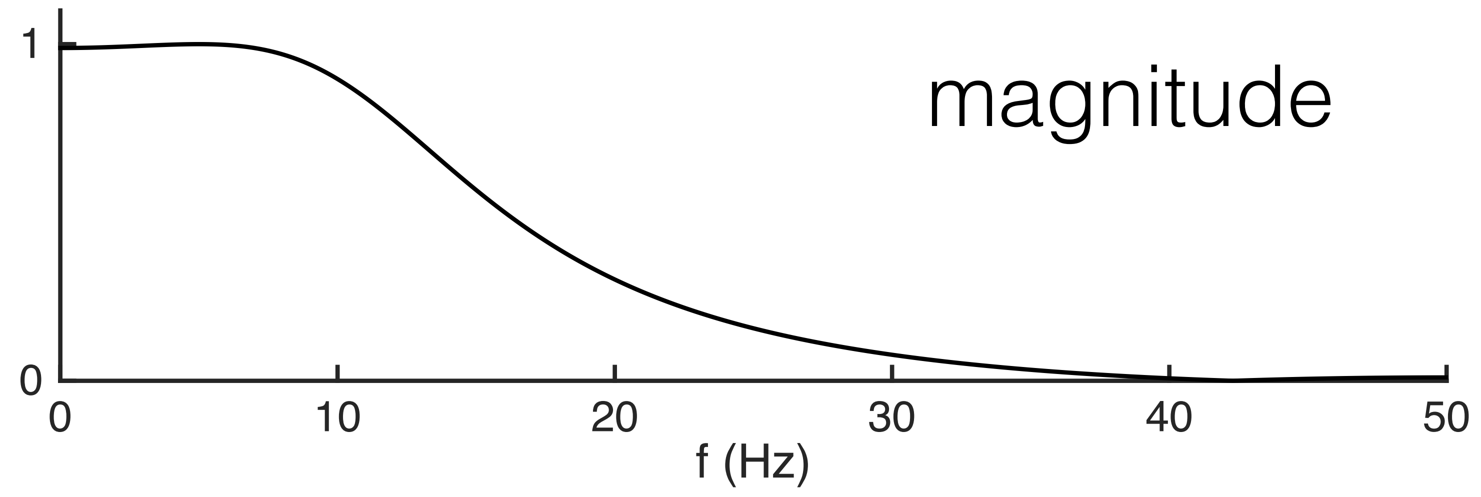
- The frequency transition is where the group delay of an IIR filter is longest.

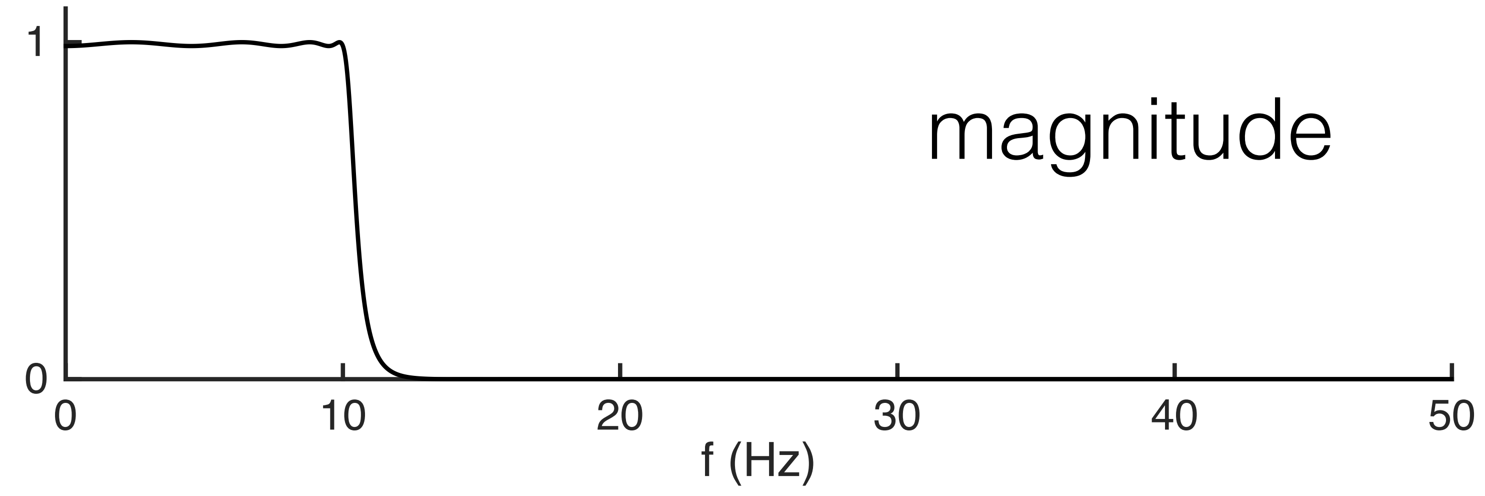




Group Delay: IIR filters

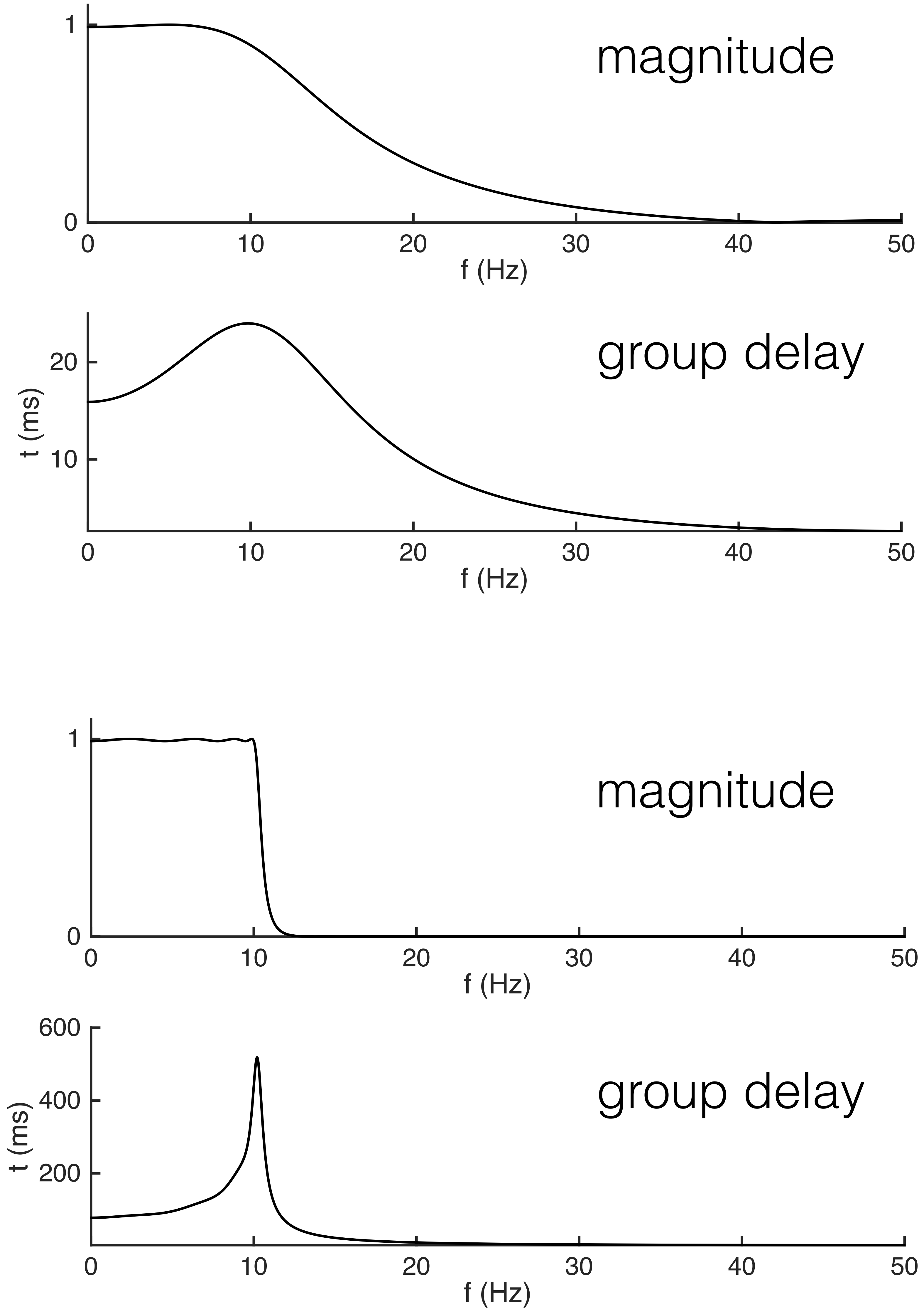
- The frequency transition is where the group delay of an IIR filter is longest.
  - The sharper the transition, the longer the group delay





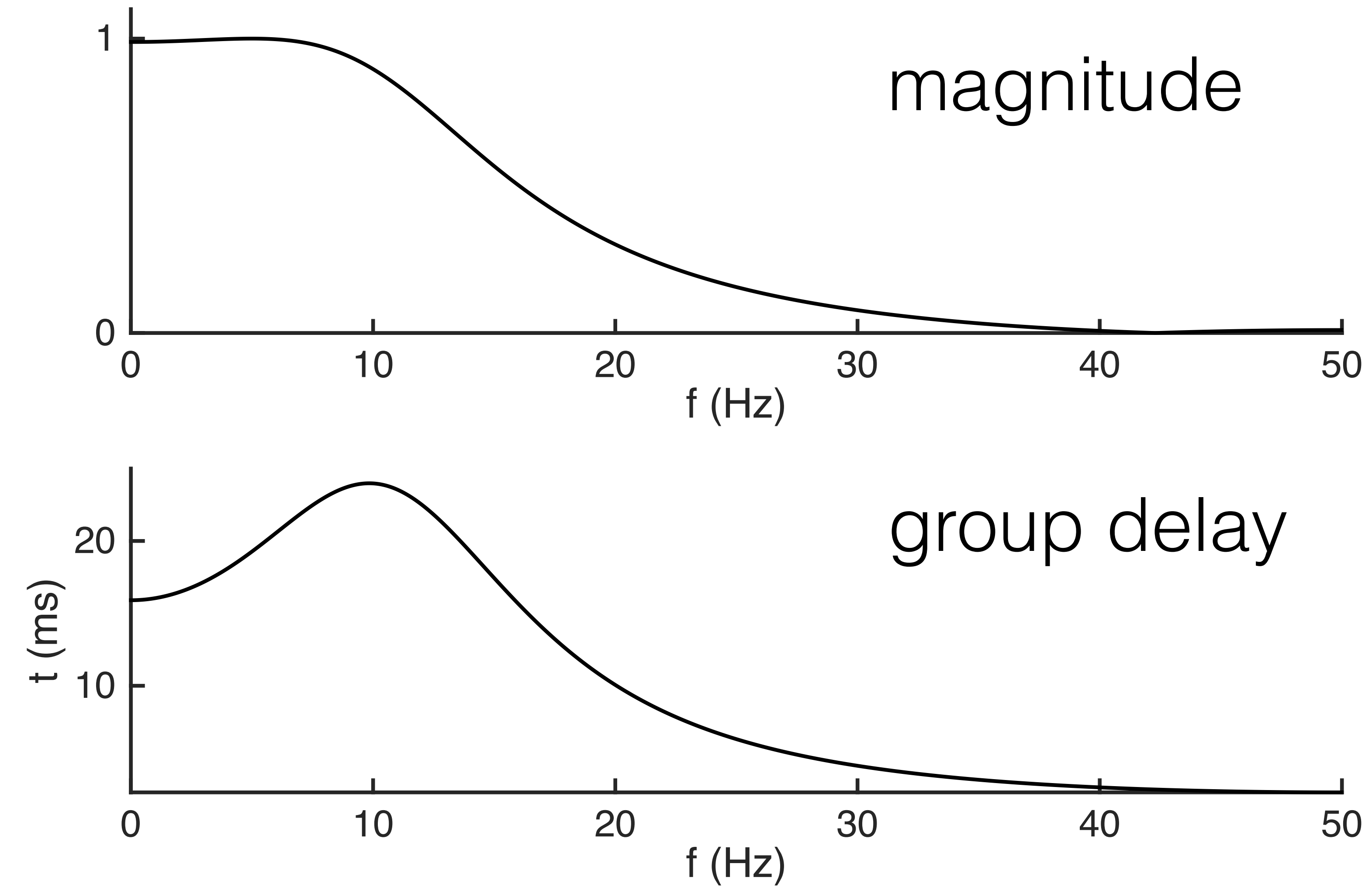
Group Delay: IIR filters

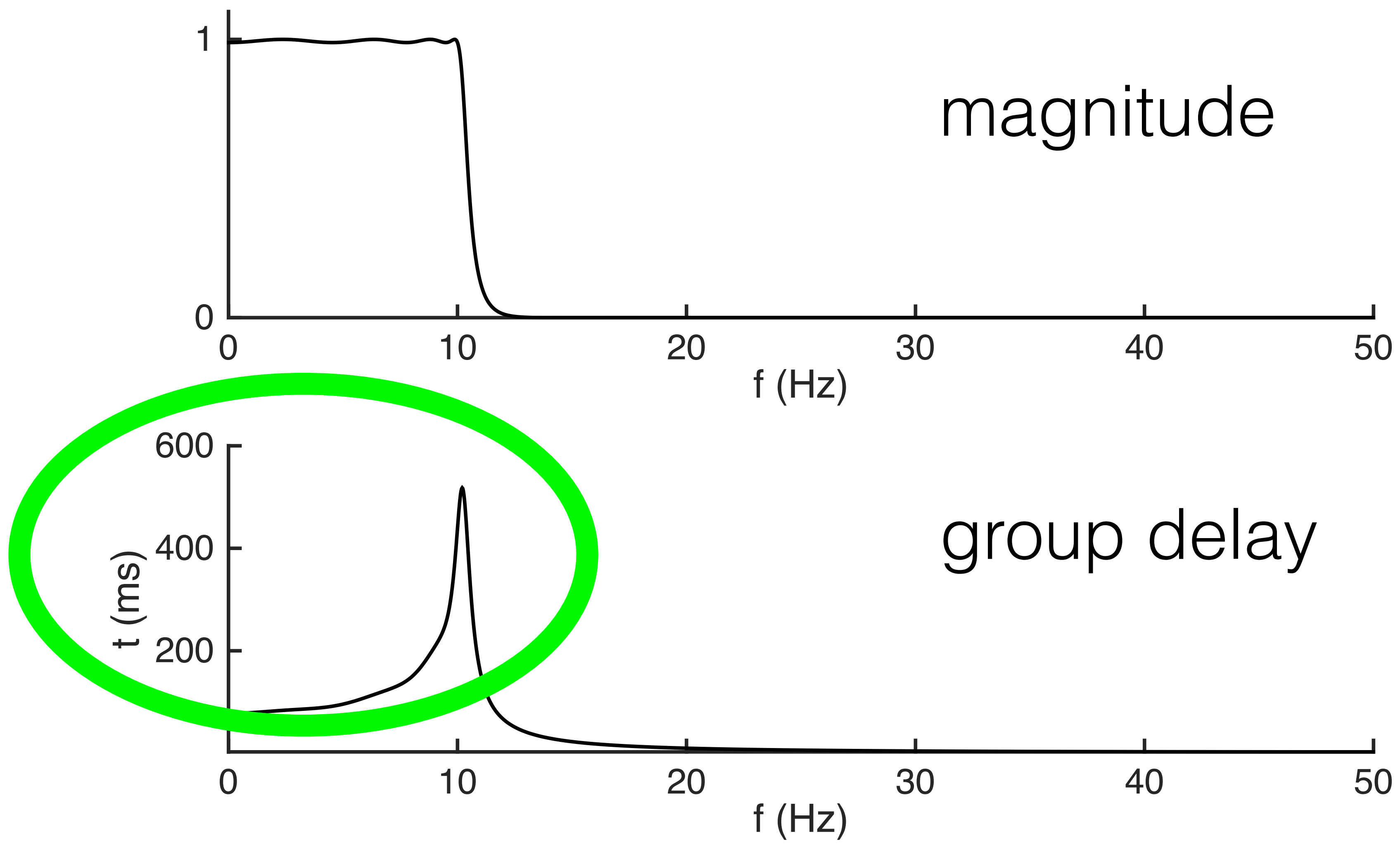
- The frequency transition is where the group delay of an IIR filter is longest.
  - The sharper the transition, the longer the group delay



Group Delay: IIR filters

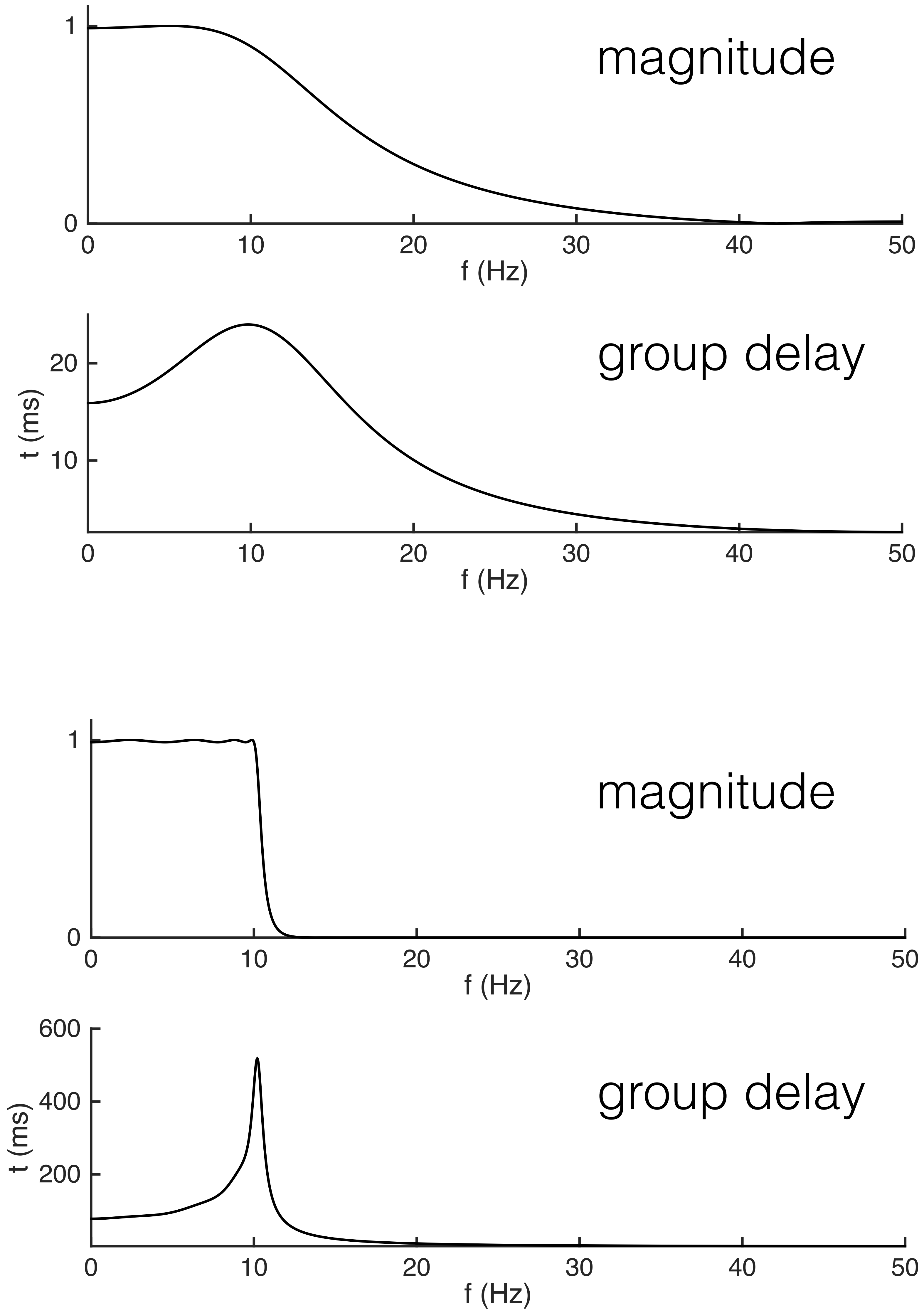
- The frequency transition is where the group delay of an IIR filter is longest.
  - The sharper the transition, the longer the group delay





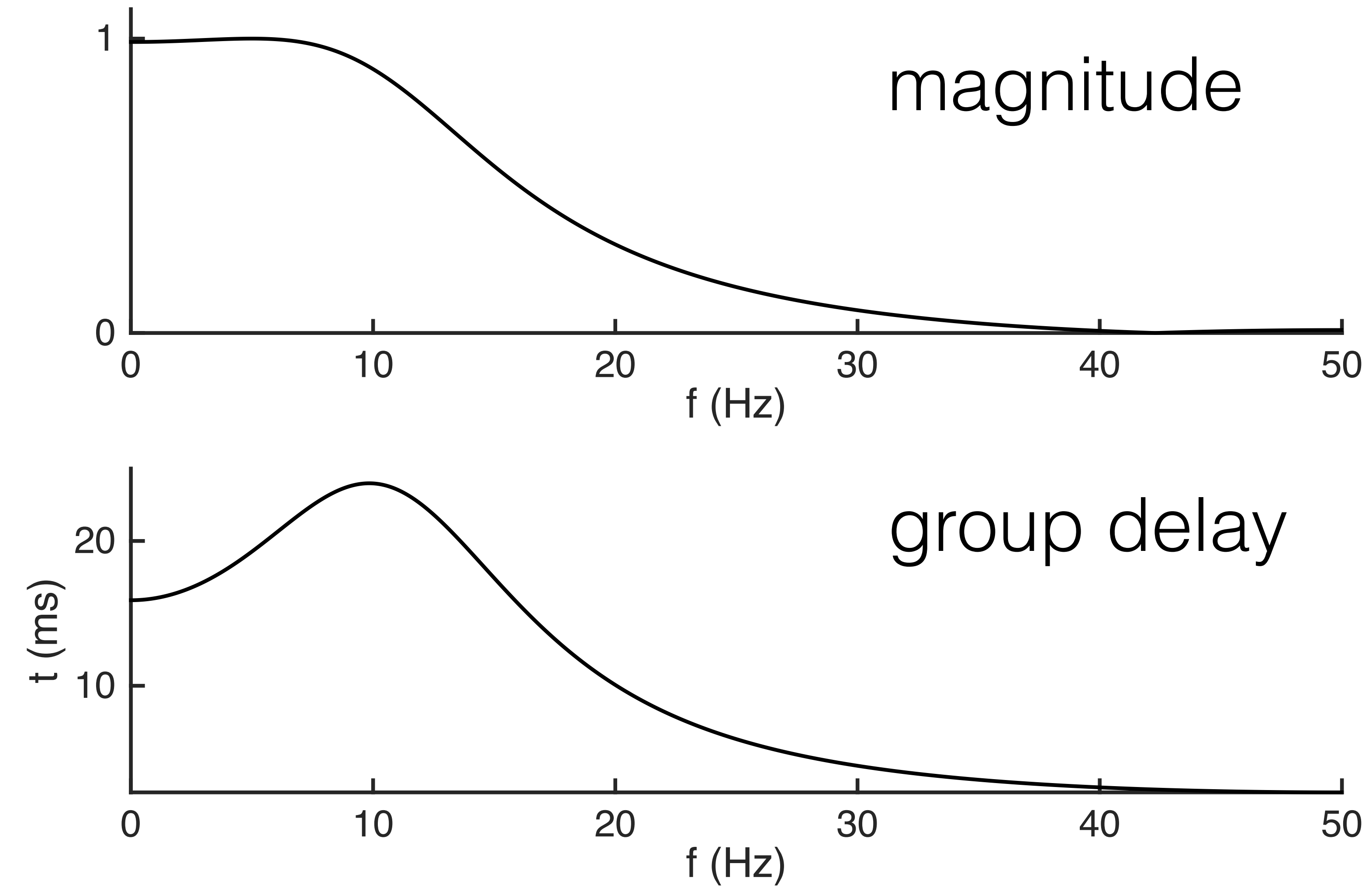
Group Delay: IIR filters

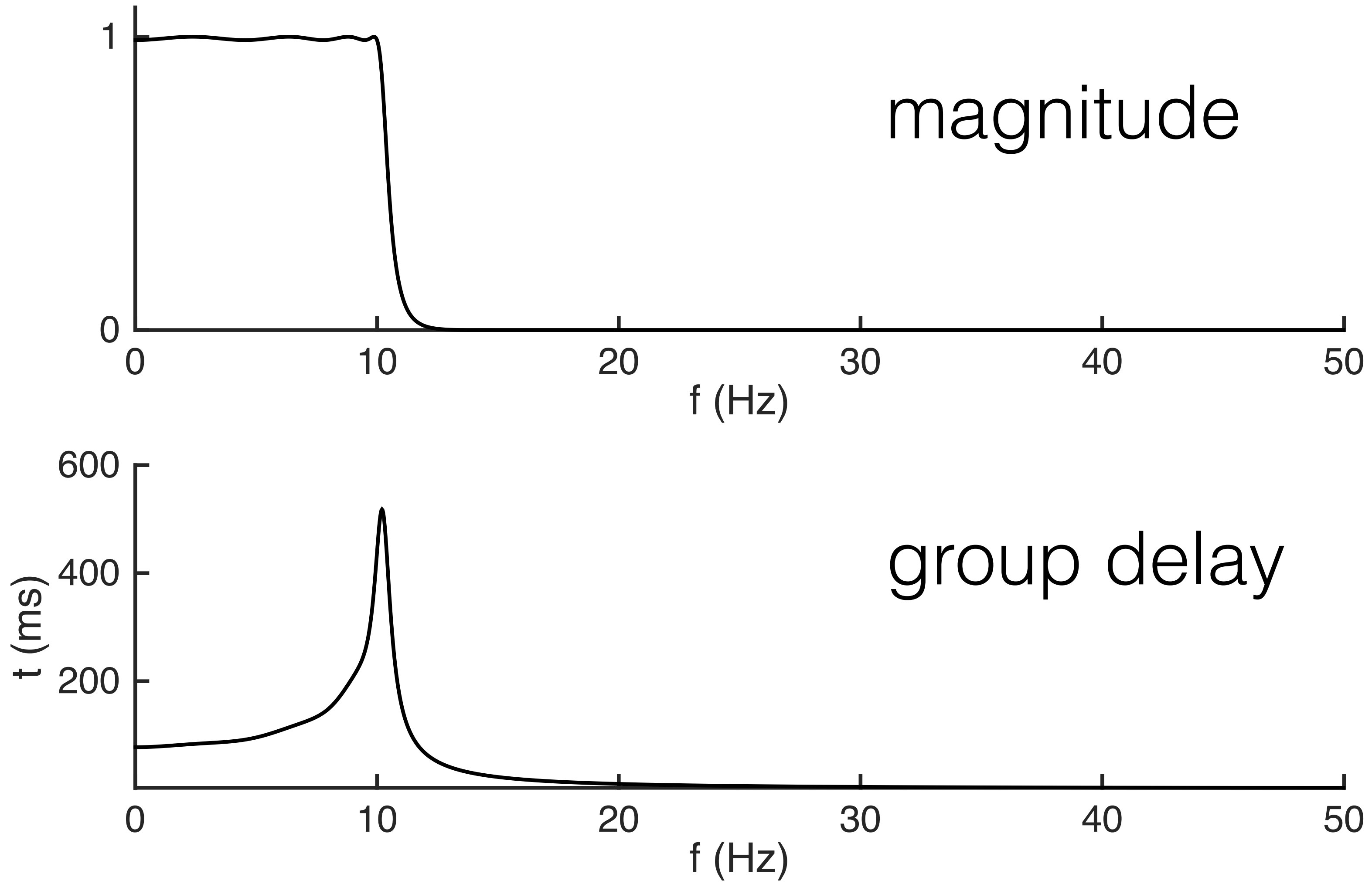
- The frequency transition is where the group delay of an IIR filter is longest.
    - The sharper the transition, the longer the group delay
    - Calculating latencies? You may need to compensate (still possible for peaks dominated by passed frequencies far from the transition).



Group Delay: IIR filters

- The frequency transition is where the group delay of an IIR filter is longest.
  - The sharper the transition, the longer the group delay
  - Calculating latencies? You may need to compensate (still possible for peaks dominated by passed frequencies far from the transition).
  - The group delay of an IIR filter does not linearly scale with Δ*t*(!), so no penalty for filtering at low sampling frequency.





Group Delay may not Matter

- For many experimental designs, only differences of latencies matter, not absolute latencies.
- For such experimental outcomes, most neural features' *shape* differences are not large.
- If both hold, separate group delays *cancel out* for latency difference.
- For such experiments, you ***may*** *not have to compensate for group delay at all*.

Signal Loss due to Filter Startup

- Output signal value depends on signal values in the past
- When calculating output at the very first moment of time, *there is no past to rely on!*
- Until filter output settles down, in time, the output signal is not well defined.

Signal Loss due to Filter Startup

For FIR filters, this problem goes away entirely after $N_{order} \times \Delta t$.

$$y[t] = \frac{1}{2}x[t] - \frac{1}{2}x[t - \Delta t]$$

Signal Loss due to Filter Startup

For FIR filters, this problem goes away entirely after $N_{order} \times \Delta t$.

$$y[t] = \frac{1}{2}x[t] - \frac{1}{2}x[t-\Delta t]: \quad y[0] = \frac{1}{2}x[0] - \frac{1}{2}x[-\Delta t]$$

Signal Loss due to Filter Startup

For FIR filters, this problem goes away entirely after $N_{order} \times \Delta t$.

$$y[t] = \frac{1}{2}x[t] - \frac{1}{2}x[t-\Delta t]: \quad y[0] = \frac{1}{2}x[0] - \frac{1}{2}x[-\Delta t] \qquad y[\Delta t] = \frac{1}{2}x[\Delta t] - \frac{1}{2}x[0]$$

Signal Loss due to Filter Startup

For FIR filters, this problem goes away entirely after $N_{order} \times \Delta t$.

$$y[t] = \frac{1}{2}x[t] - \frac{1}{2}x[t-\Delta t]: \quad y[0] = \frac{1}{2}x[0] - \frac{1}{2}x[-\Delta t] \qquad y[\Delta t] = \frac{1}{2}x[\Delta t] - \frac{1}{2}x[0]$$

- Recommendation: either keep extra *earlier* data of duration $N_{order} \times \Delta t$, or prepend the same amount of zero signal (Matlab's default). Consider this "warmup" time for the filter. Then toss out this same amount from the output.
- This works well for small N_{order}.
- This is another reason to use FIR filters only of low order.
- This is another reason FIR filters may work best at high sample rates.

Signal Loss due to Filter Startup

For IIR filters, the problem is more subtle

$$y[t] = \frac{1}{10} x[t] - \frac{9}{10} y[t - \Delta t]$$

Signal Loss due to Filter Startup

For IIR filters, the problem is more subtle

$$y[t] = \frac{1}{10}x[t] - \frac{9}{10}y[t-\Delta t]: \qquad y[0] = \frac{1}{10}x[0] - \frac{9}{10}y[-\Delta t]$$

Signal Loss due to Filter Startup

For IIR filters, the problem is more subtle

$$y[t] = \frac{1}{10}x[t] - \frac{9}{10}y[t-\Delta t]: \qquad y[0] = \frac{1}{10}x[0] - \frac{9}{10}y[-\Delta t] \qquad y[\Delta t] = \frac{1}{10}x[\Delta t] - \frac{9}{10}y[0]$$

Signal Loss due to Filter Startup

For IIR filters, the problem is more subtle

$$y[t] = \frac{1}{10}x[t] - \frac{9}{10}y[t-\Delta t]: \qquad y[0] = \frac{1}{10}x[0] - \frac{9}{10}y[-\Delta t] \qquad y[\Delta t] = \frac{1}{10}x[\Delta t] - \frac{9}{10}y[0]$$

- The output depends not only on the input in the past, but also on the filter output of the past.

- Recommendation: again keep extra earlier data (warmup time), *as much you can afford*. Then toss out the same amount from the output.

- If keeping enough earlier data not feasible, Matlab permits supplying pre t=0 initial data. Using this with reasonable values can really help shrink warmup time.

    - Even prepending data from the *end* of the signal may help over nothing.

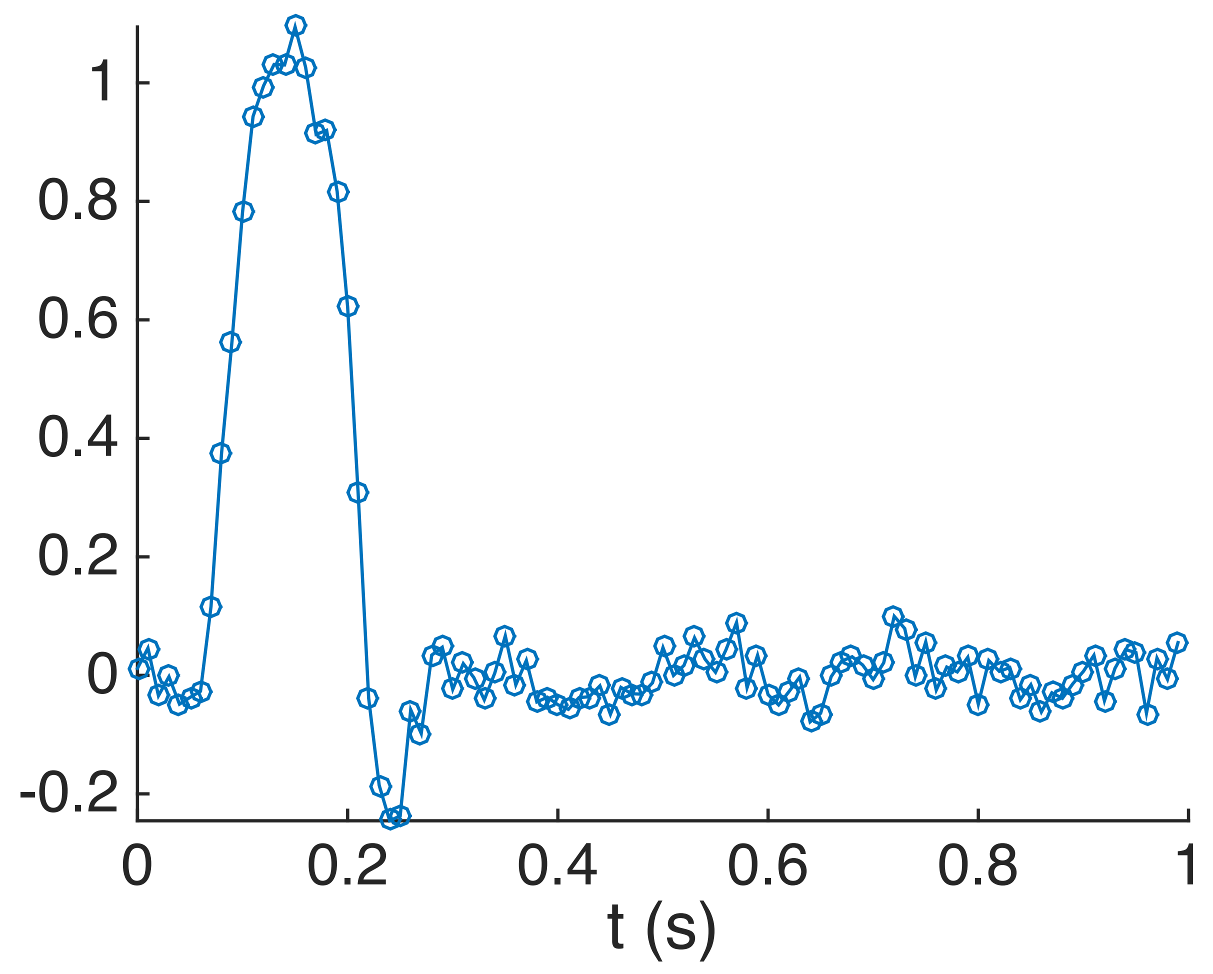
Stability concerns for IIR filters

- IIR filters employ feedback; might be negative (good) or positive (bad)
- Common IIR filters designed to be stable: all feedback negative (good)
- Design can break down due to numerical roundoff error
- Breakdown more likely for higher order filters
- Recommendation: only use low order ($N_{order} < 10$) IIR filters.
  - Lower order IIR filters also have less sharp frequency transitions, so this is rarely a burden.

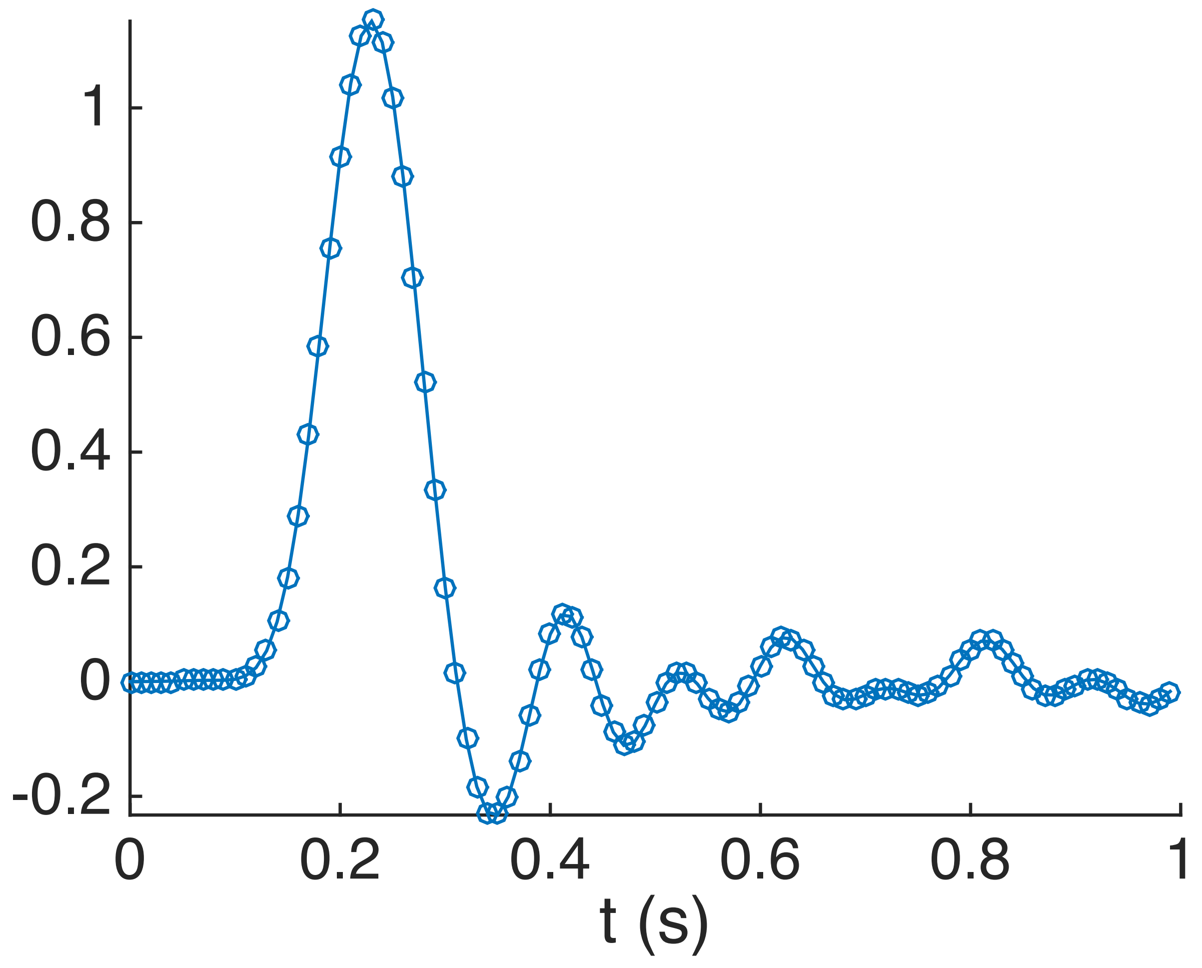
How Would I even Notice Instability?

It’s not subtle (but only if you know where to look)

Raw Signal



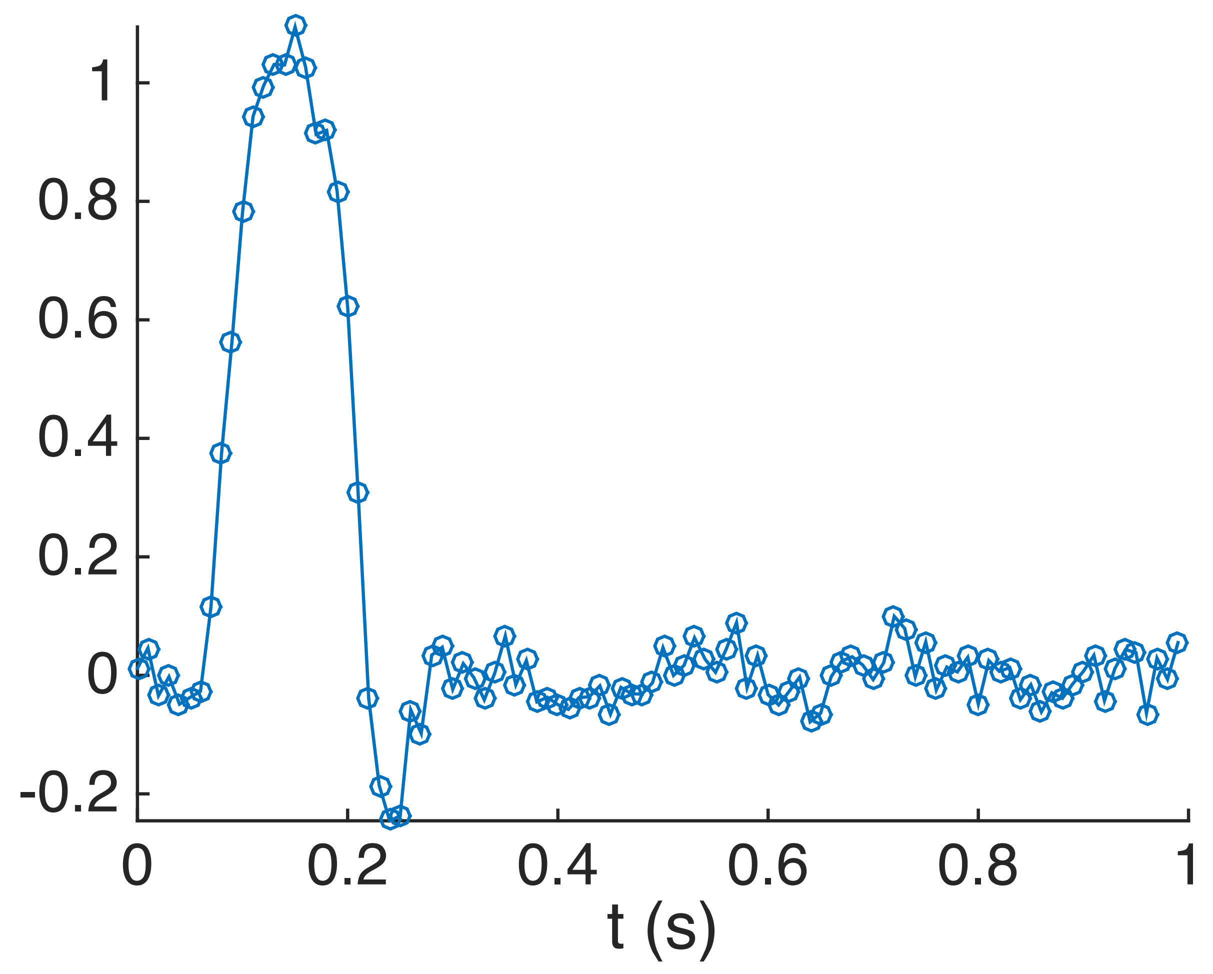
Stable Filter



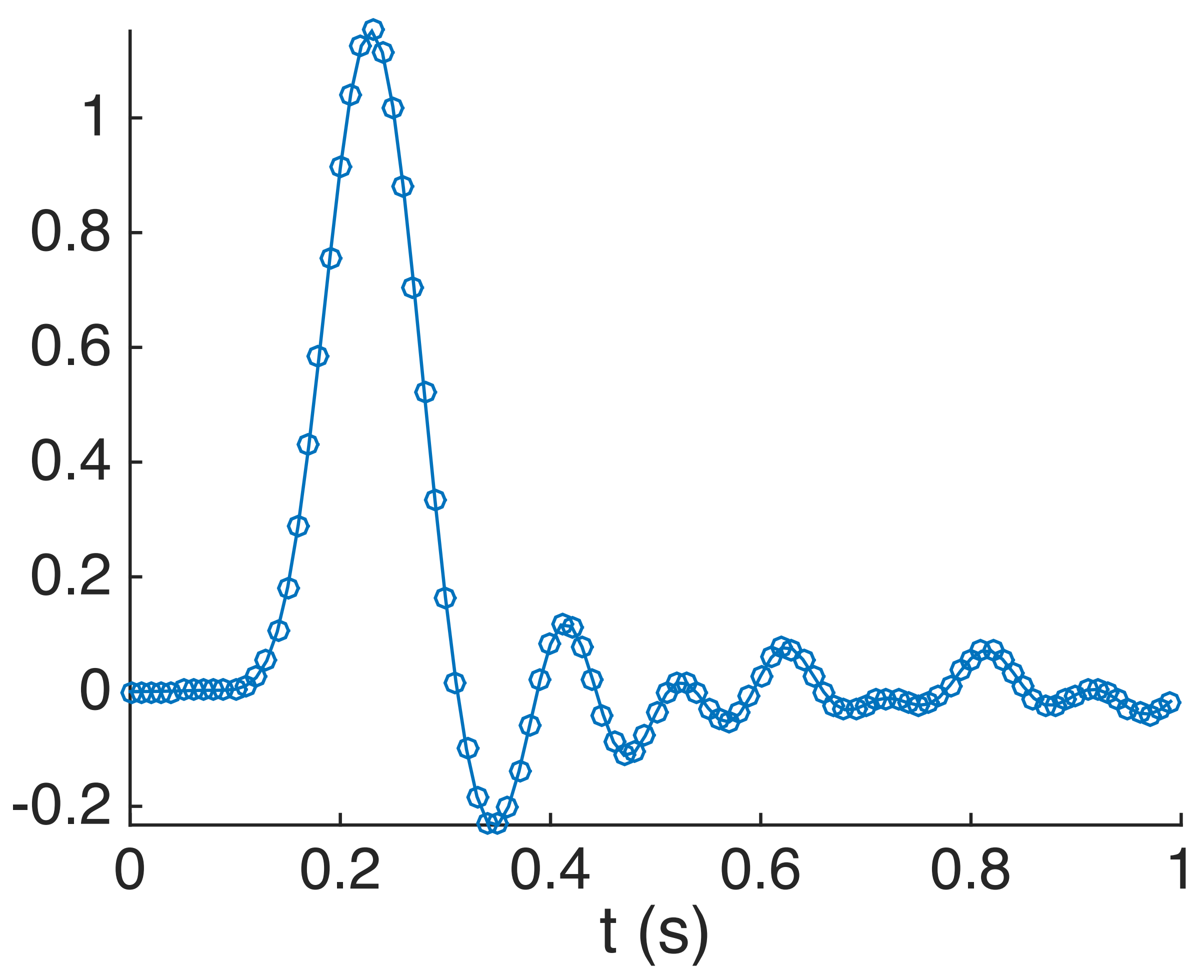
How Would I even Notice Instability?

It’s not subtle (but only if you know where to look)

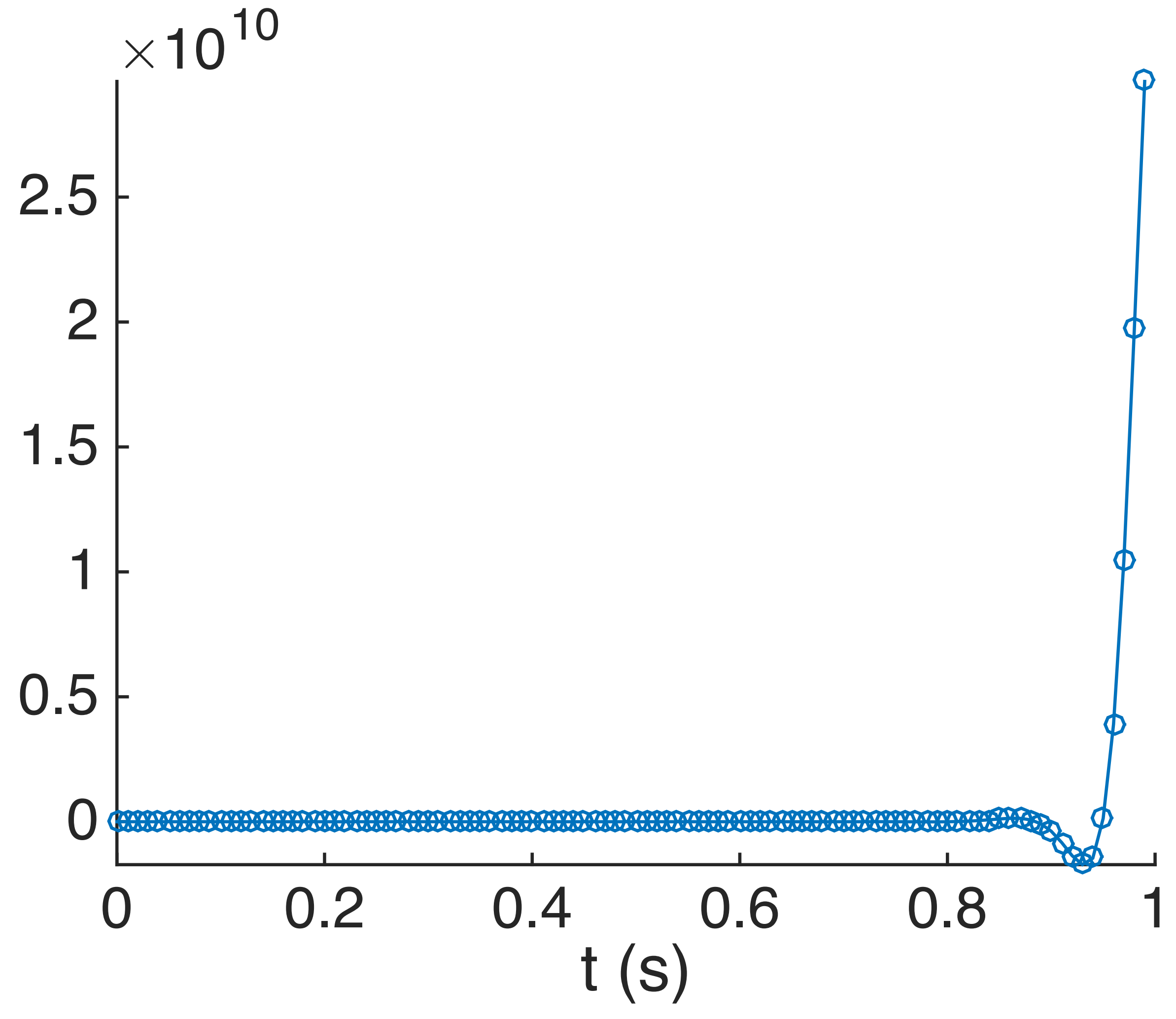
Raw Signal



Stable Filter



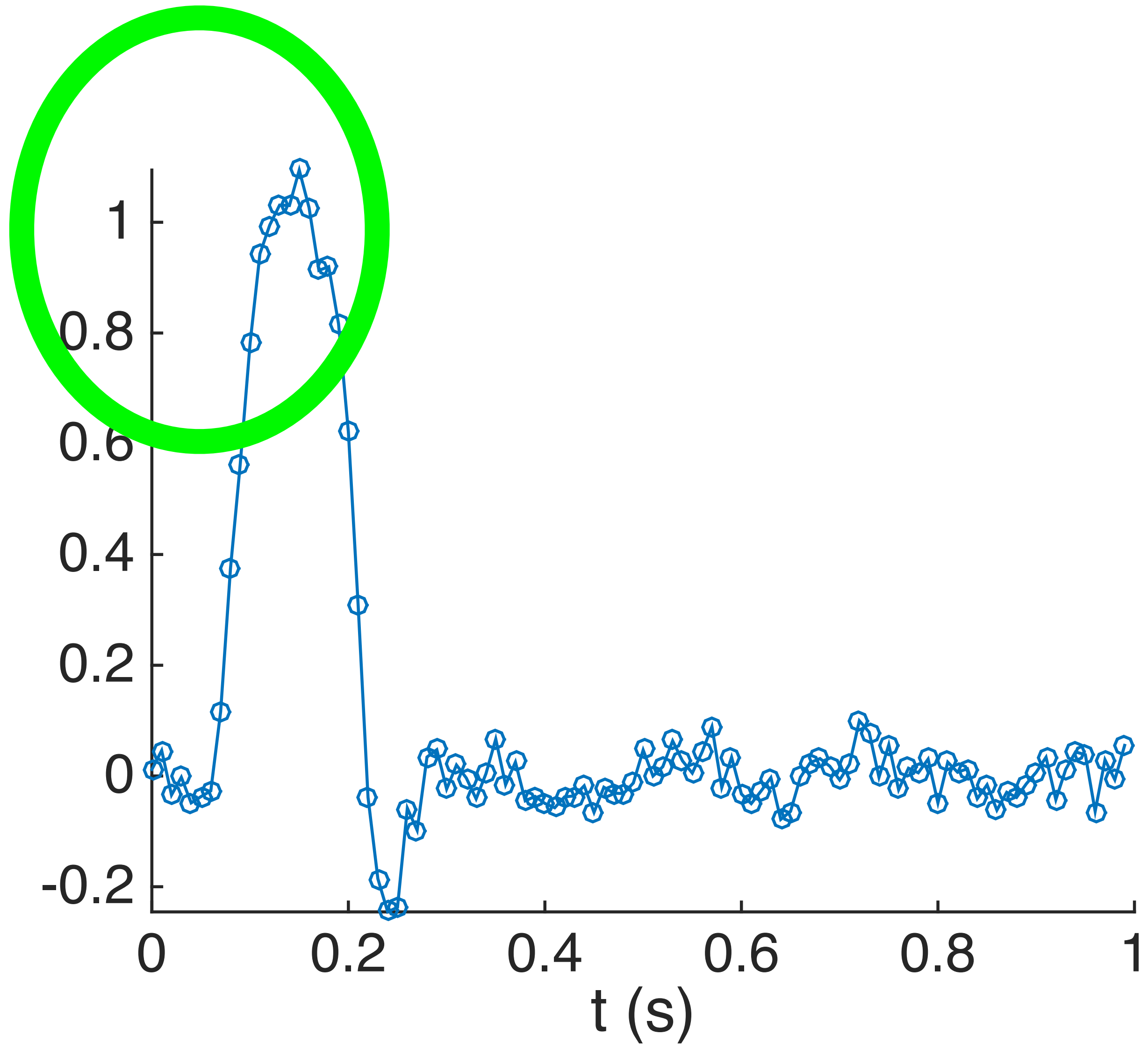
Filter gone Unstable



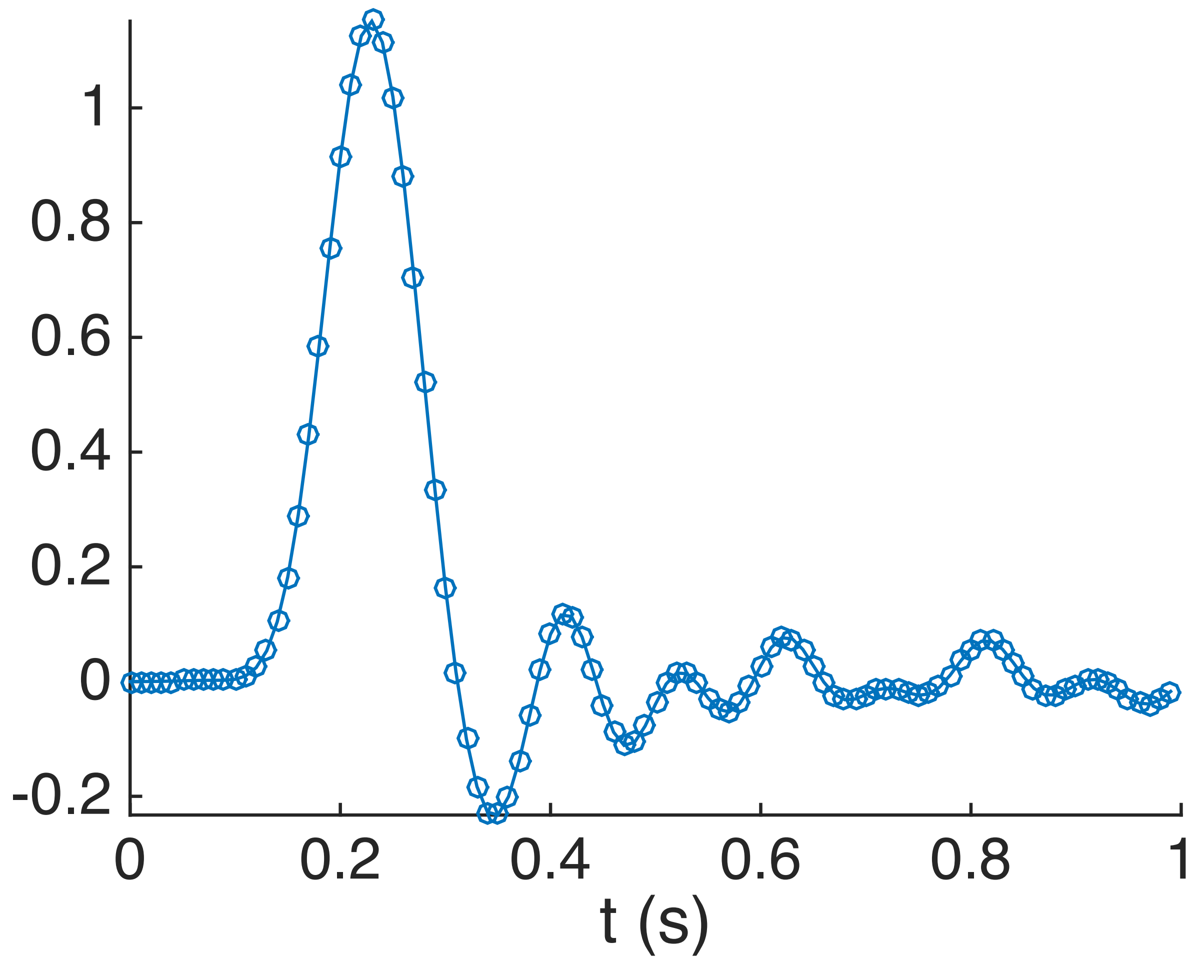
How Would I even Notice Instability?

It’s not subtle (but only if you know where to look)

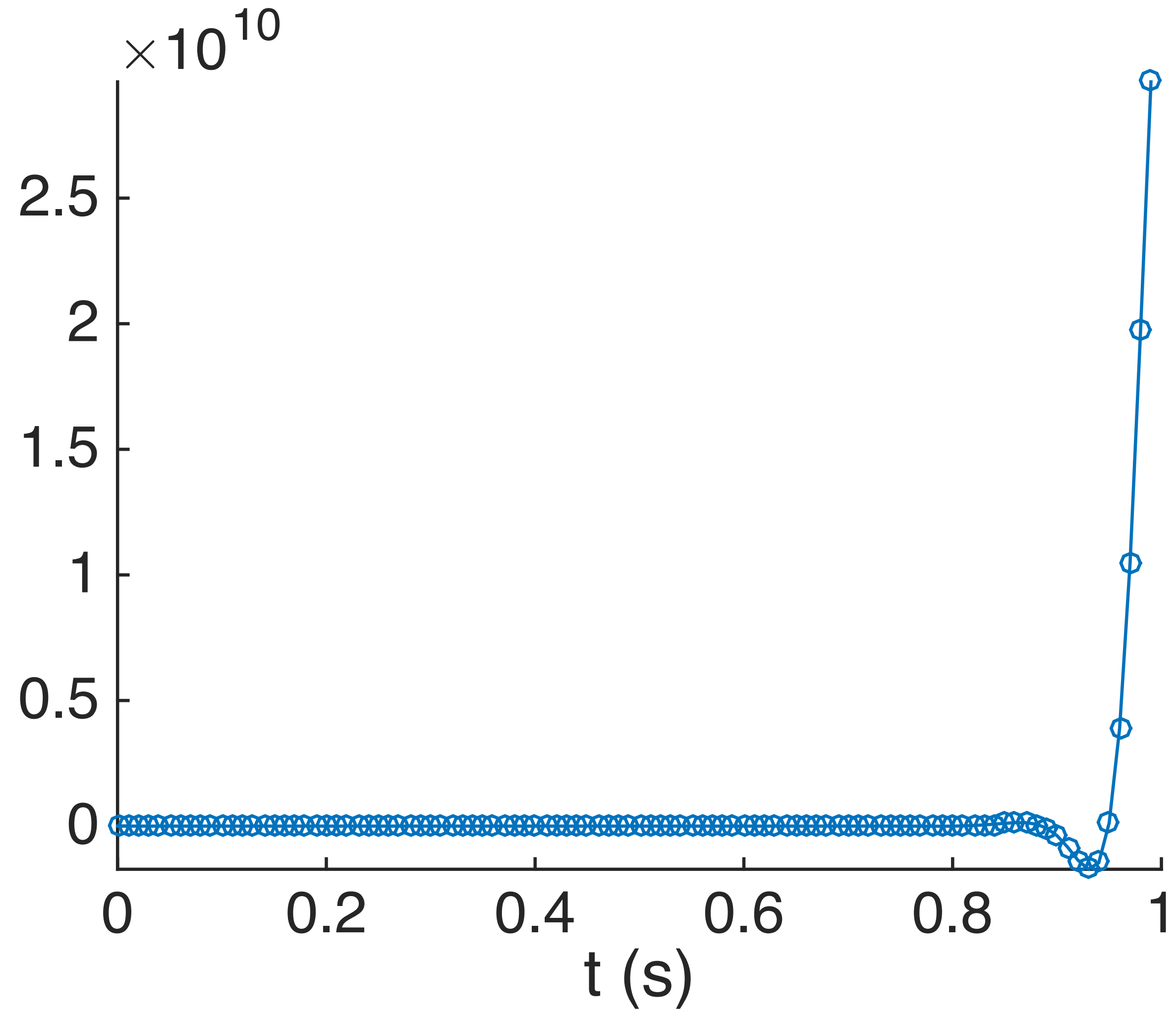
Raw Signal



Stable Filter

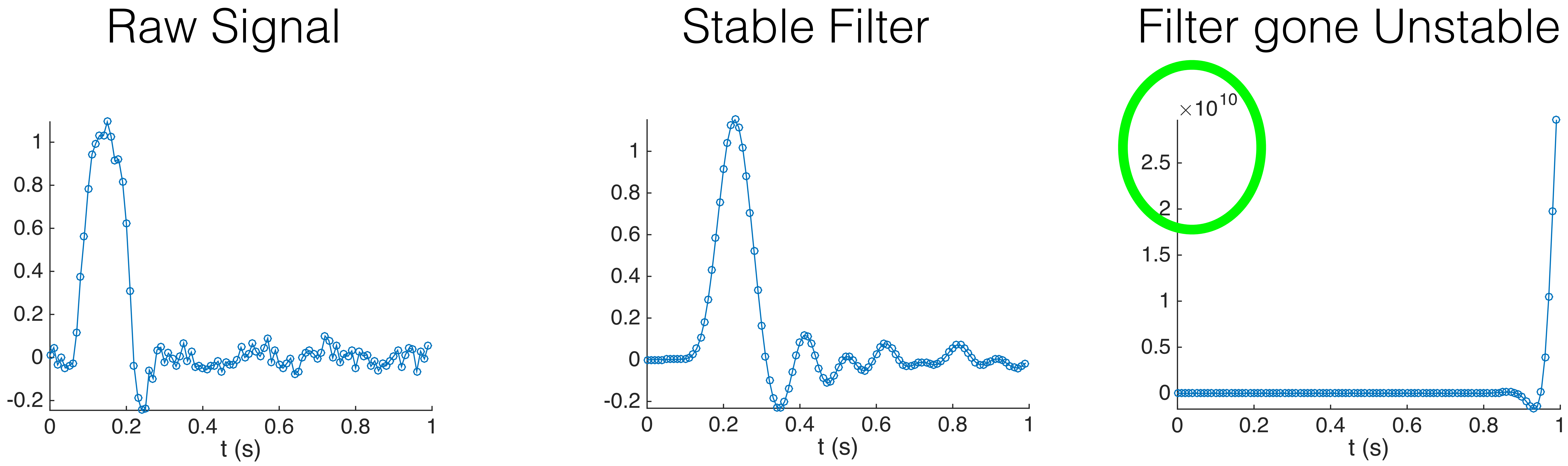


Filter gone Unstable



How Would I even Notice Instability?

It’s not subtle (but only if you know where to look)



How Do I Choose a Filter?

For high sampling frequency and plenty of initial data, consider FIR filters

- This is typically appropriate for raw, un-epoched data.

- Parks-McClellen ("optimal") filters work well. Can choose soft frequency transitions.

  - (Report the filter choice and order, as well as all cutoff frequencies and any other specified parameters, in your *Methods* section.)

- Take care with software "black-box" FIR filters. Maybe good, maybe not.

  - *How much quality signal processing does the software author know?*

How Do I Choose a Filter?

Otherwise, consider IIR filters

- This is typically appropriate for epoched data.

- If can’t be bothered, Butterworth filters are “fine”.
    - If really can’t be bothered, use a 4th order Butterworth.
- If you care about your frequency bands, consider using an Elliptic filter.
    - (Report the filter choice and order, as well as all cutoff frequencies and any other specified parameters, in the *Methods* section.)
- Software “Black-box” IIR filters usually not worrisome, even if not optimal for data.

How Do I Choose a Filter?

If you care about your frequency bands, consider using an Elliptic filter

- Needs “slop” factors/tolerances
  - In the *pass* frequency band, how close to “1” (100% passes through) do you *really* need? If your peak height were off by only 1%, would you even notice?
    - Matlab requires this (“passband ripple”) to be in dB: 1% ≈ 0.1 dB
  - In the *stop* frequency band, how close to “0” (0% passes through) do you *really* need? If your noise is suppressed only by 100x (not infinitely), would you notice?
    - Matlab requires this (“stopband attenuation”) to be in dB: 100x = 40 dB

Outline

- Fourier Transform: *Why It's Useful*, and *What it Can/Cannot Do For You*
- Filters: *What They Do*, and *How They Do It*
- Filters: *Why So Many Different Kinds? Which Should I Use and When*?
- Grab Bag:
  - *Use Causal Filters; Windowing is Good*

Outline

- Fourier Transform: *Why It's Useful*, and *What it Can/Cannot Do For You*
- Filters: *What They Do*, and *How They Do It*
- Filters: *Why So Many Different Kinds? Which Should I Use and When*?
- Grab Bag:
    - *Use Causal Filters; Windowing is Good*

Grab Bag

- Use Causal Filters
- Windowing is Good

Causal & non-Causal Filtering

All filters discussed hear are *causal.*

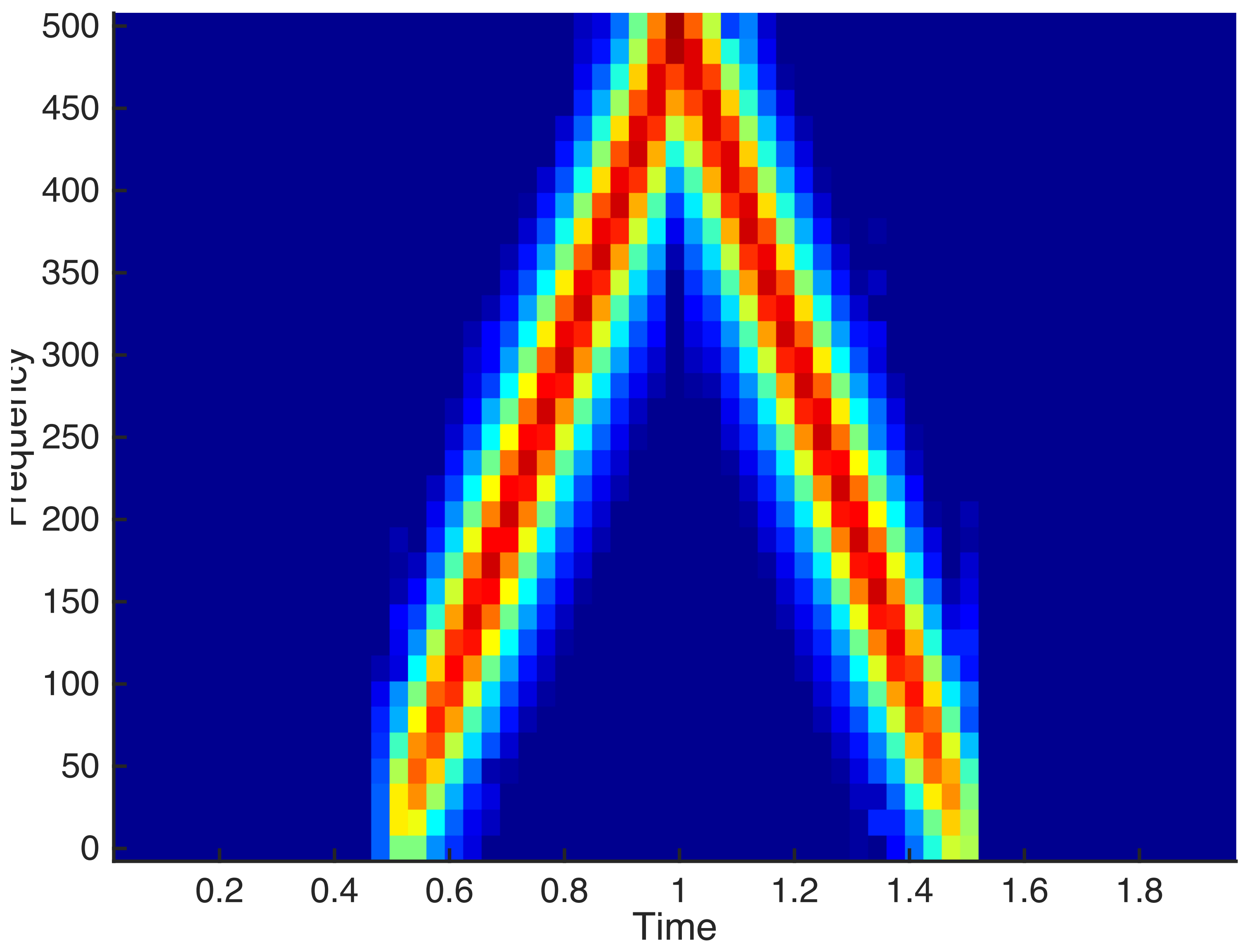
- Variation in the input signal causes variation in the output. The output variation occurs at the same instant as in the input, *or, most likely, later*, but never earlier: Lengthening/Delay is normal.
- Some types of lengthening are desirable: using a low pass filter to slow down fast changes in the input signal.
- Some types of lengthening are undesirable: ringing due to sharp frequency transition.

Causal & non-Causal Filtering

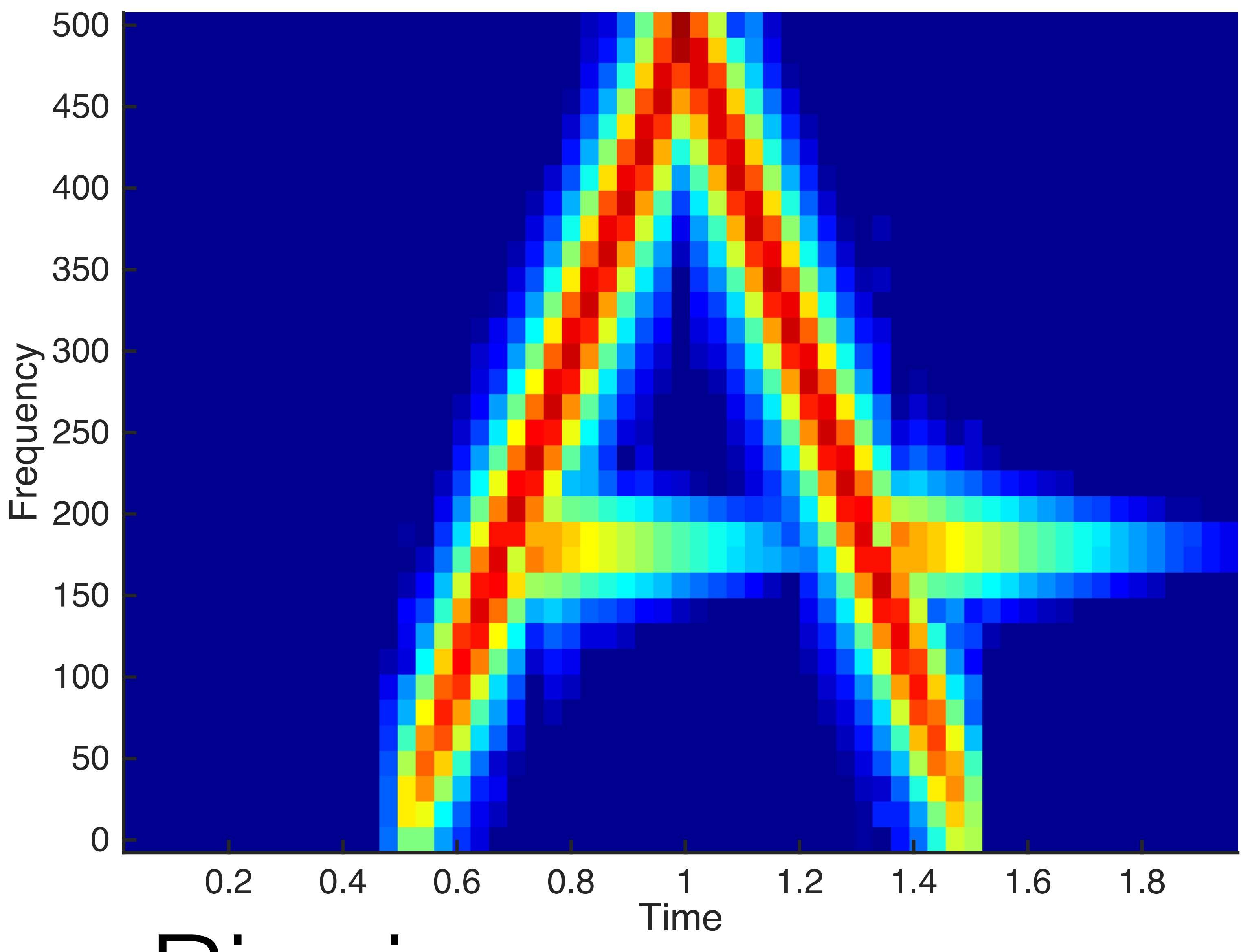
- It is mathematically possible (but biologically undesirable!), to temporally "center" all such output changes so they do not seem to be all contribute to delay.
- This (undesirable act) can be achieved with a particular kind of non-causal filtering: *zero-phase* filtering (Matlab "`filtfilt`").
- Zero-phase sounds wonderful, but it is **not** (c.f. "ideal" filter).
- Zero-phase filters *do not remove delay-based artifacts*, and in fact they double them.

Zero-Phase Filtering Example

FM Sweep
(Spectrogram)



Notched FM Sweep

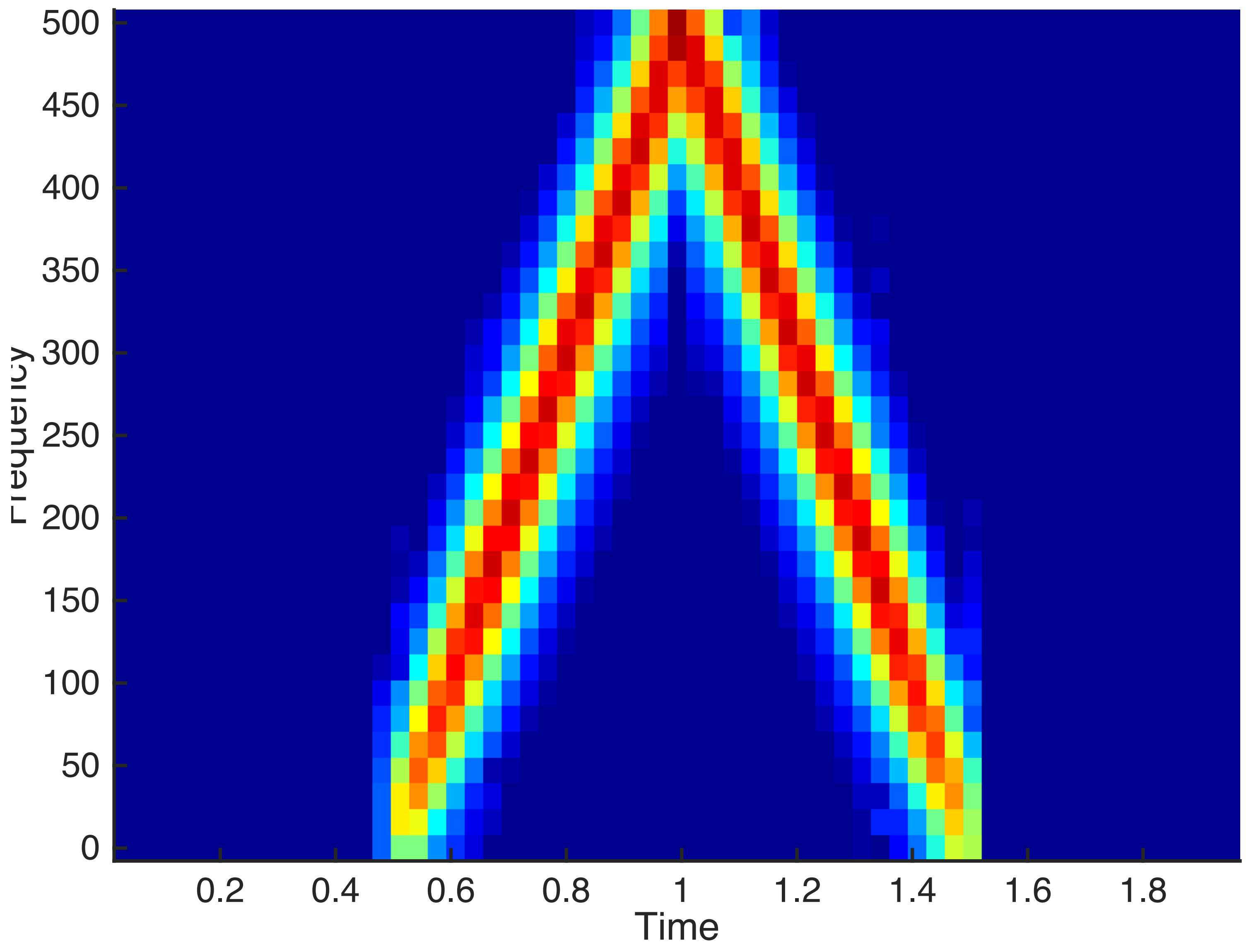


Ringing:

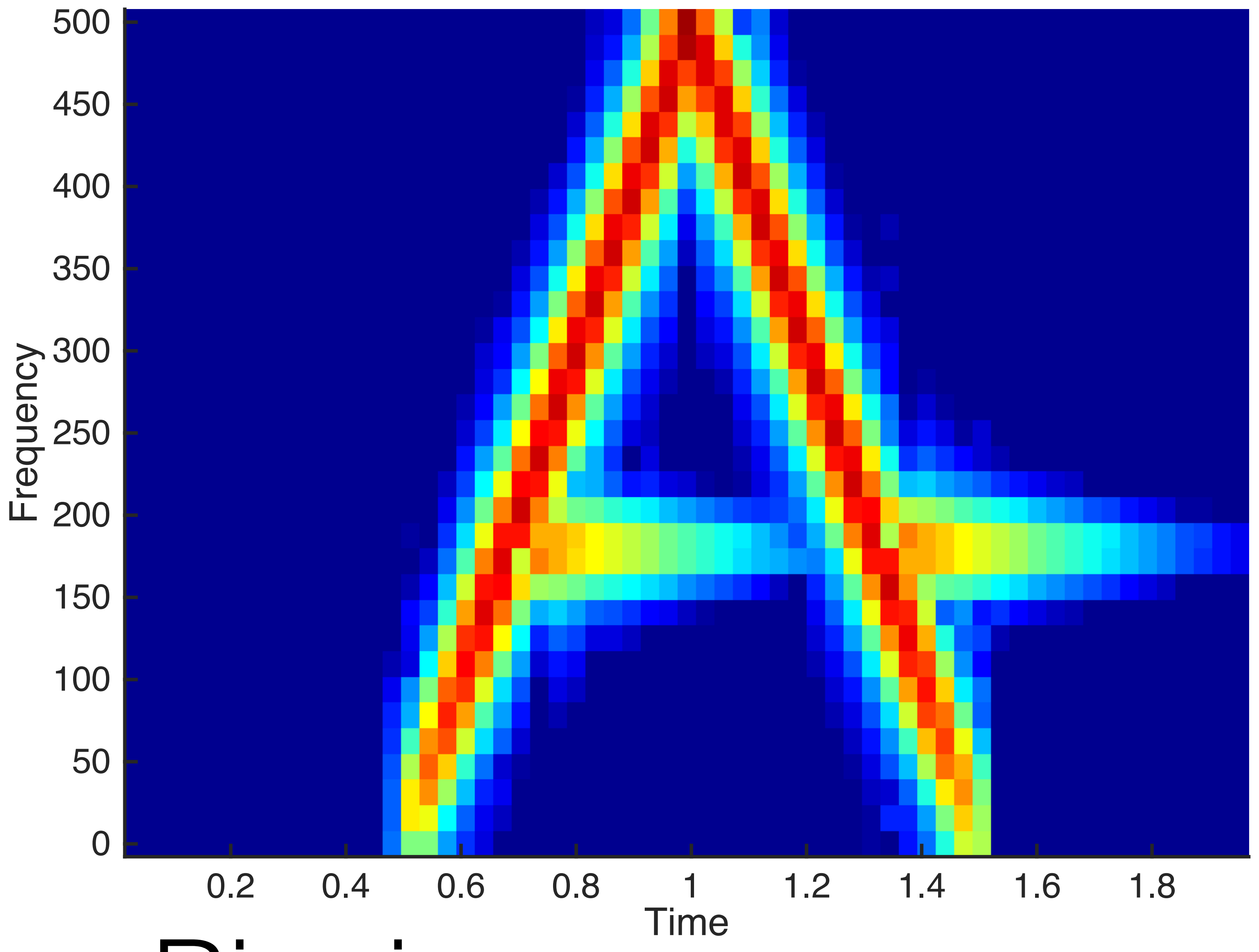
- persistent
- causal

Zero-Phase Filtering Example

FM Sweep
(Spectrogram)



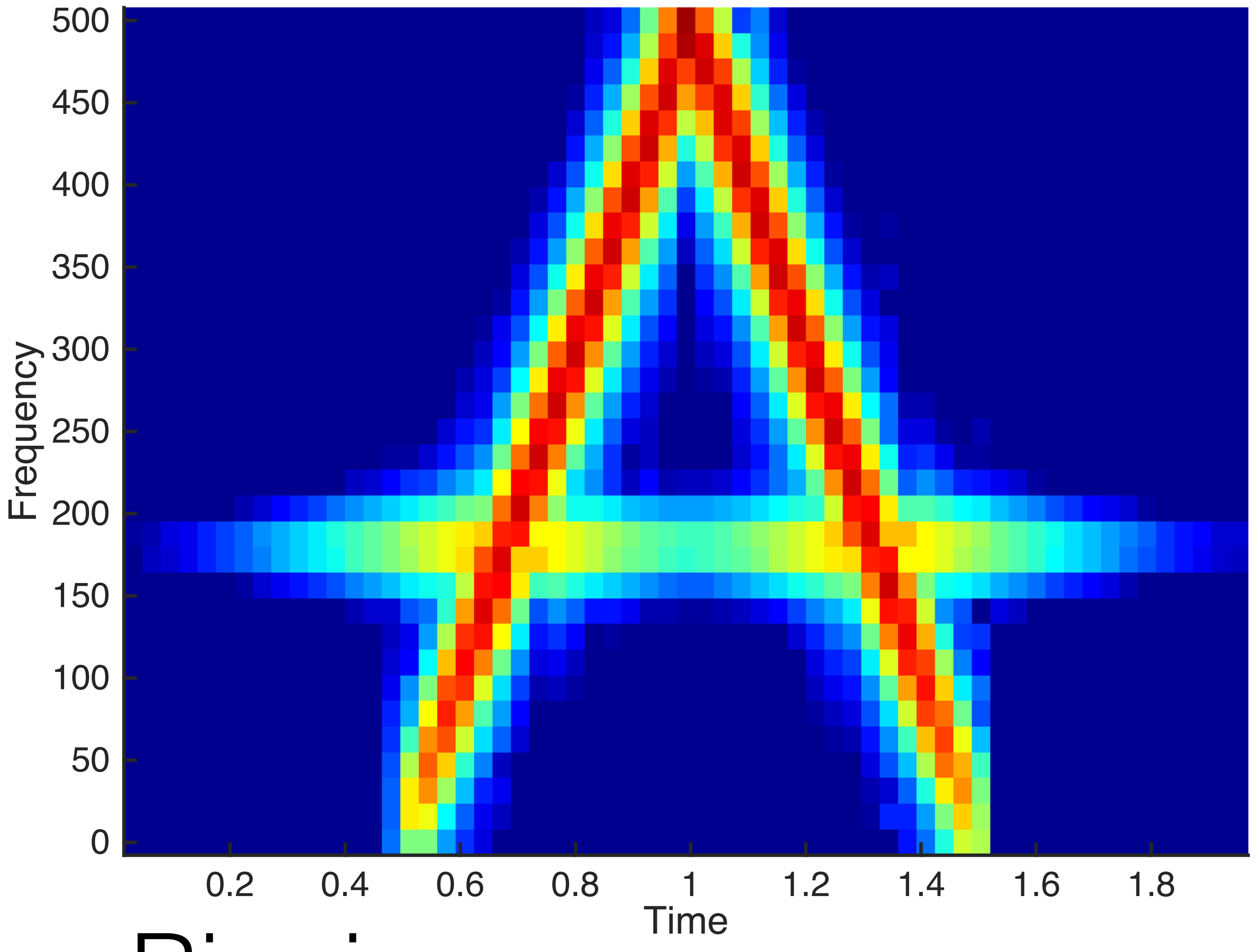
Notched FM Sweep



Ringing:

- persistent
- causal

Zero-Phase Notched
FM Sweep



Ringing:

- duplicated and flipped
- no cancellation (except "on average"?)

Causal & non-Causal Filtering

- Zero-phase filters do not remove distortions, but instead *replicate* them *backwards in time* (symmetrically, if signals are ~symmetric).

- Replicating them backwards may give zero "on average" *but not actually zero*.

- *Large, Longer Latency* neural features (e.g., motor system responses) can be *artificially shifted backwards in time*(!).

    - **Detection/Decision event may be contaminated with future Motor responses.**

- Compensation for delayed feature-peak may even be OK, *but be very careful about other features*: not-actually-delayed *rise-to-peak* replaced with pre-causal rise-to-peak.

- Recommendation: Don't use. Causes more problems than solves.

Break for Computer Lab Exercise 6

- Zero-phase filters do not remove distortions, but instead *replicate* them *backwards in time* (symmetrically, if signals are ~symmetric).

- Replicating them backwards may give zero "on average" *but not actually zero*.

- *Large, Longer Latency* neural features (e.g., motor system responses) can be *artificially shifted backwards in time*(!).

    - **Detection/Decision event may be contaminated with future Motor responses.**

- Compensation for delayed feature-peak may even be OK, *but be very careful about other features*: not-actually-delayed *rise-to-peak* replaced with pre-causal rise-to-peak.

- Recommendation: Don't use. Causes more problems than solves.

Grab Bag

- Use Causal Filters
- Windowing is Good

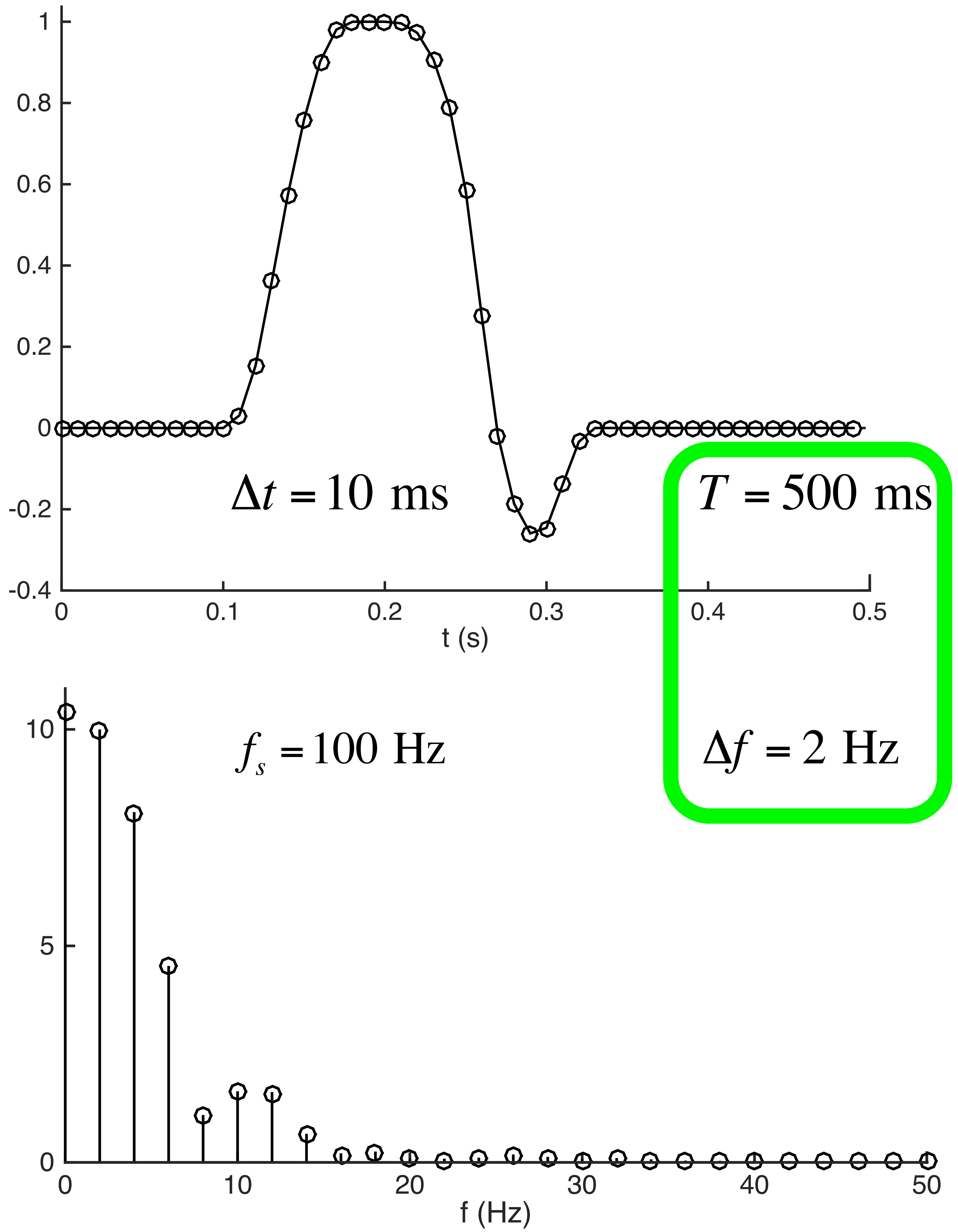
“Fourier coefficients do not always mean what you think they mean.”

–The Princess Bride (paraphrased)

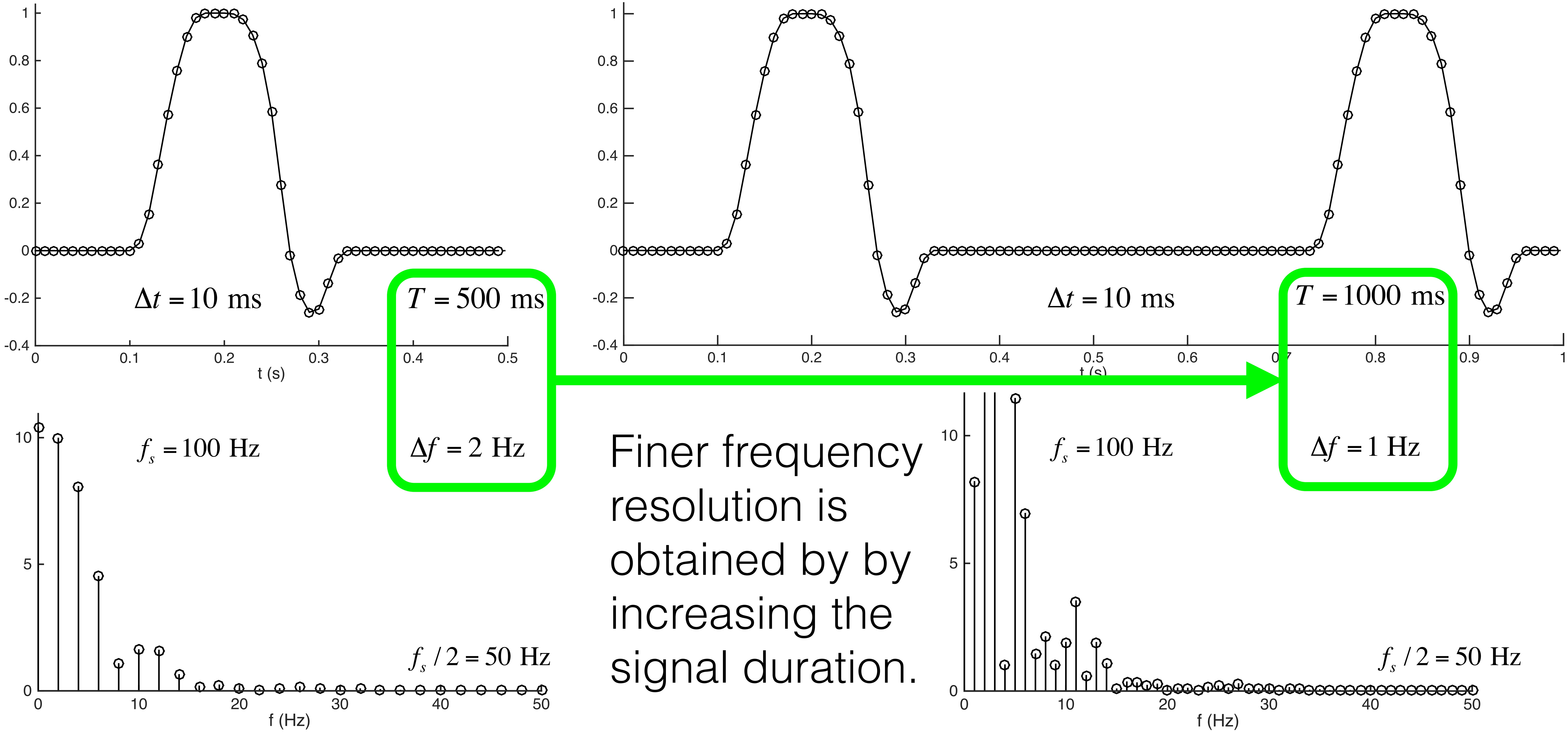
Windowing and Frequency Resolution

- *Frequency resolution* (Δf), the limiting factor in distinguishing one frequency from another, is determined by the total *duration* of the signal (T).

- This relationship is the time-frequency conjugate of the relationship between and *temporal resolution* (Δt) and *sampling frequency* (f_s).



Windowing and Frequency Resolution



Finer frequency resolution is obtained by by increasing the signal duration.

Windowing and Frequency Resolution

- It is sometimes desirable to “smear” information *temporally* (e.g. low-pass filter in order to attenuate noise).
  - The *effective* time resolution is worse, even though Δt remains unchanged.
- **Analogously**, it is sometimes desirable to “smear” information over frequencies (e.g. *power spectral density* estimation or *spectral leakage* minimization).
  - The *effective* frequency resolution becomes worse, even though Δf remains unchanged.
- This frequency smearing is typically accomplished by *windowing* in the time domain.

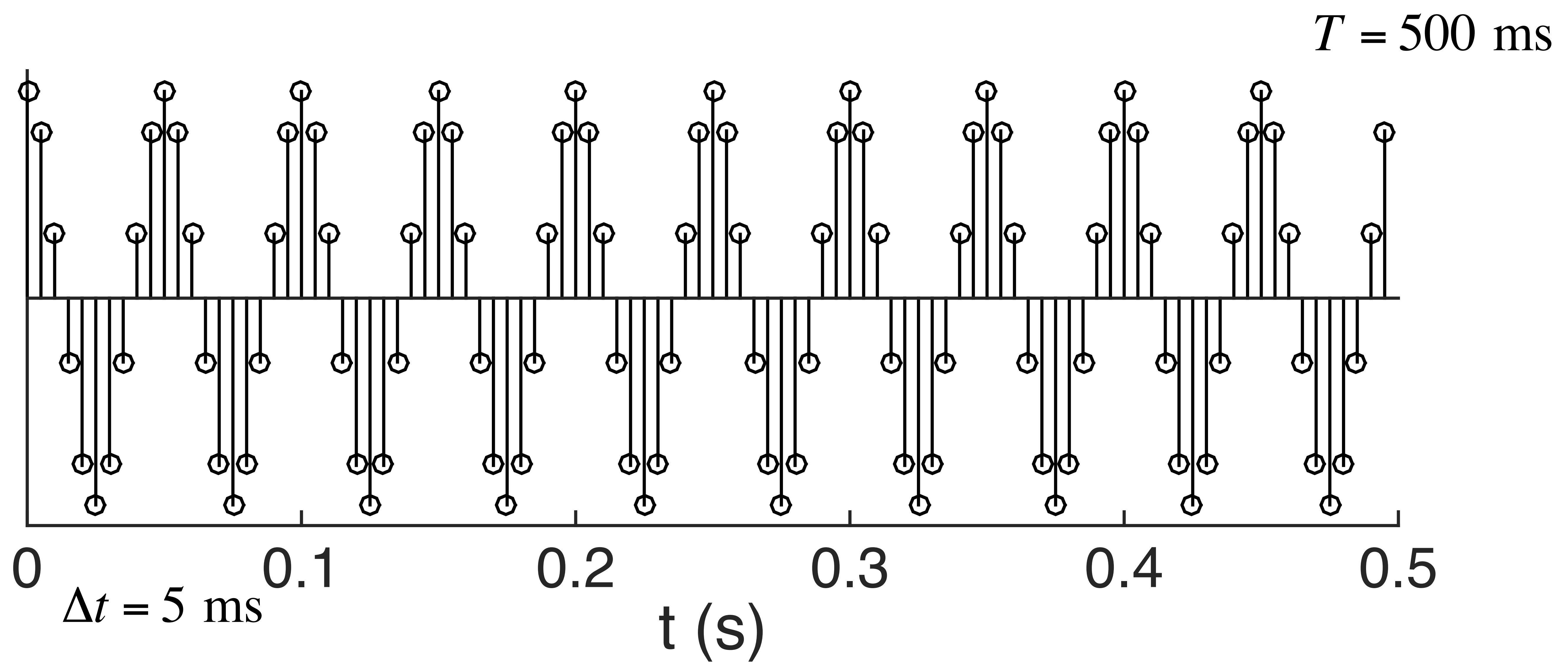
Spectral Leakage

Example 1

A pure sinusoid (single frequency).

In the Fourier domain it has a single Fourier component.

$$x[t] = \cos(2\pi f_a t)$$
$$f_a = 20 \text{ Hz}$$



Spectral Leakage

Example 2

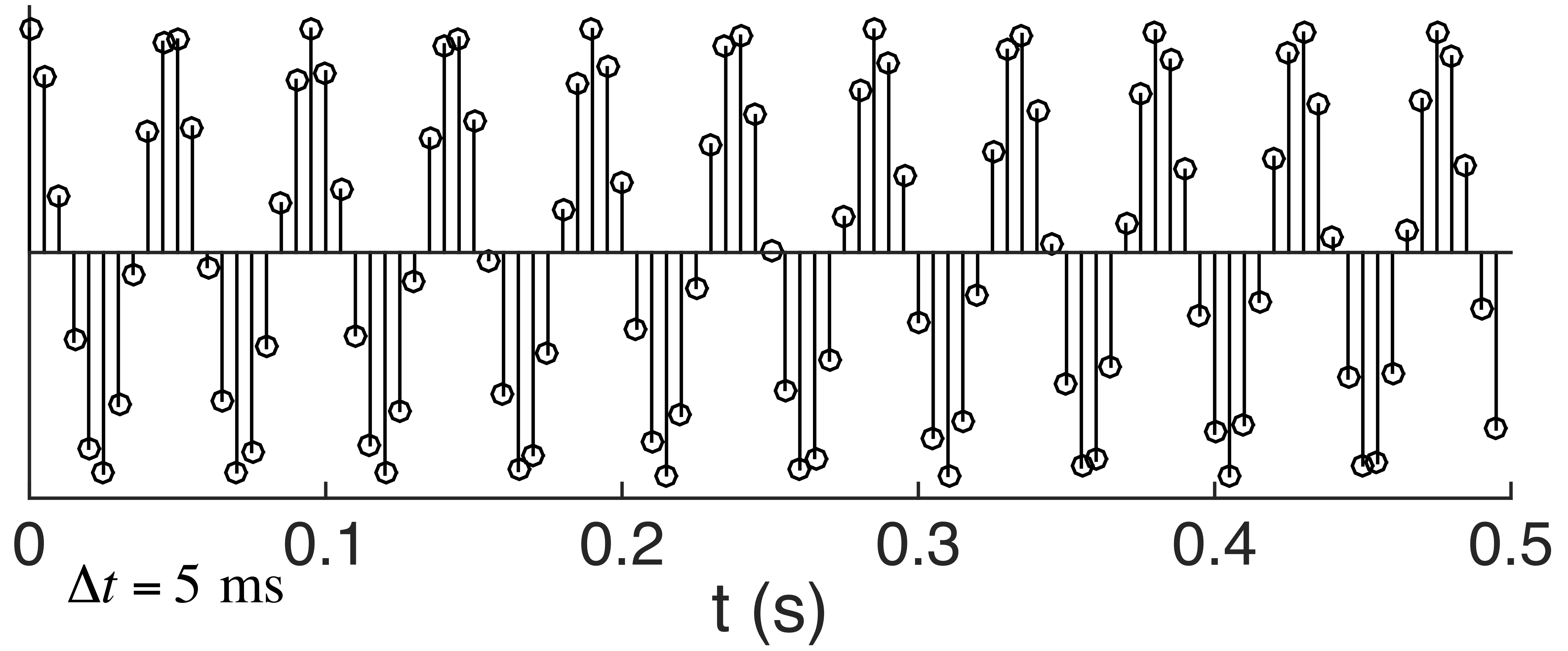
A pure sinusoid (single frequency).

What does it look like in the Fourier Domain?

$$x[t] = \cos(2\pi f_b t)$$

$$f_b = 21 \text{ Hz}$$

$T = 500$ ms



$\Delta t = 5$ ms

$f_s = 200$ Hz

$\Delta f = 2$ Hz

$f_s / 2 = 100$ Hz

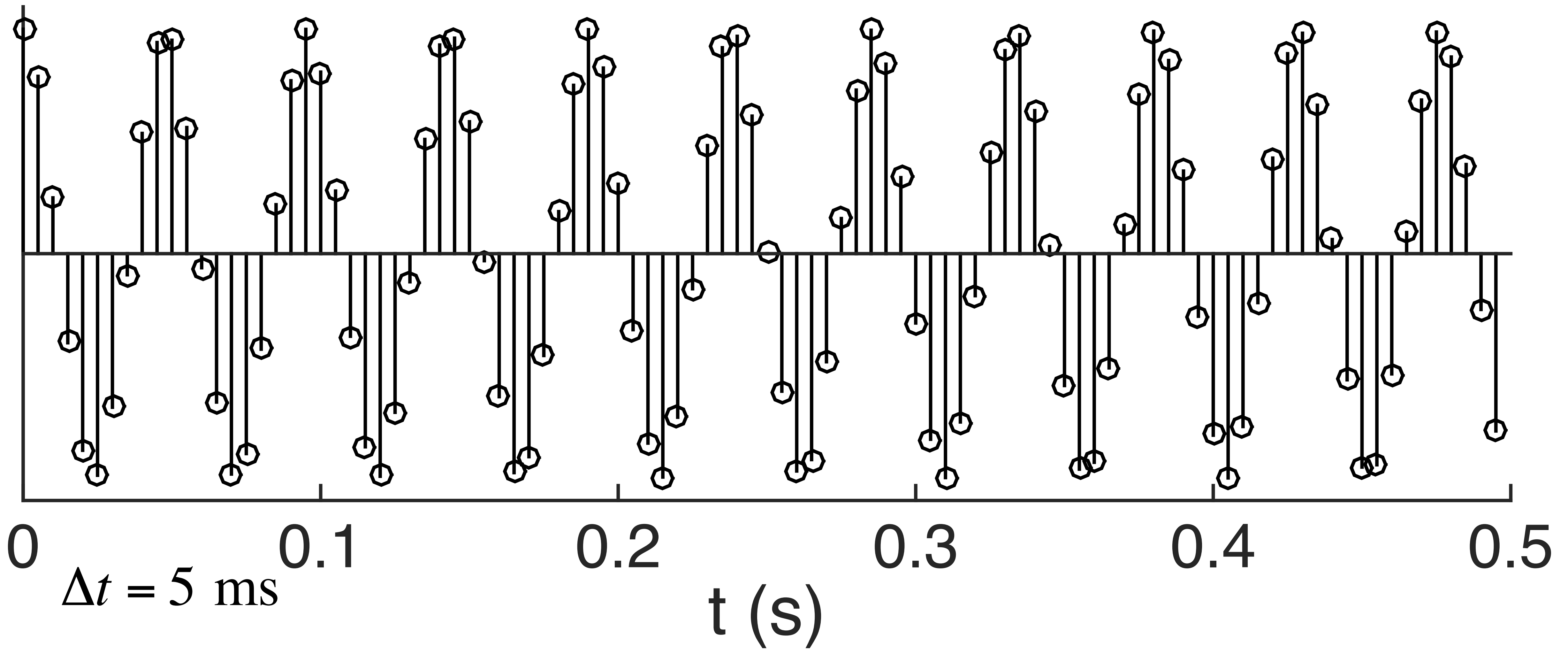
Spectral Leakage

Example 2

A pure sinusoid (single frequency).

What does it look like in the Fourier Domain?

$T = 500 \text{ ms}$



$\Delta t = 5 \text{ ms}$

$f_s = 200 \text{ Hz}$

$$x[t] = \cos(2\pi f_b t)$$

$$f_b = 21 \text{ Hz}$$

$$\Delta f = 2 \text{ Hz}$$

$\Delta f = 2 \text{ Hz}$

$f_s / 2 = 100 \text{ Hz}$

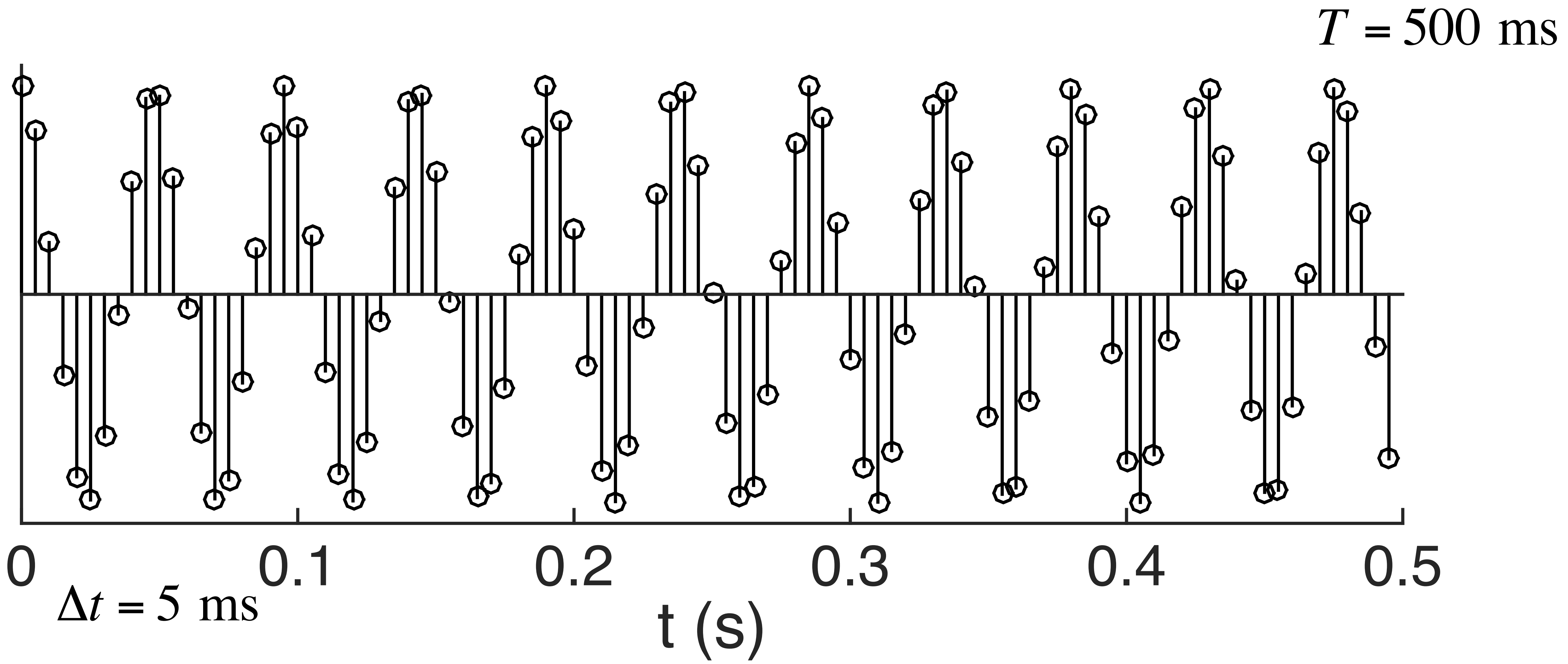
Spectral Leakage

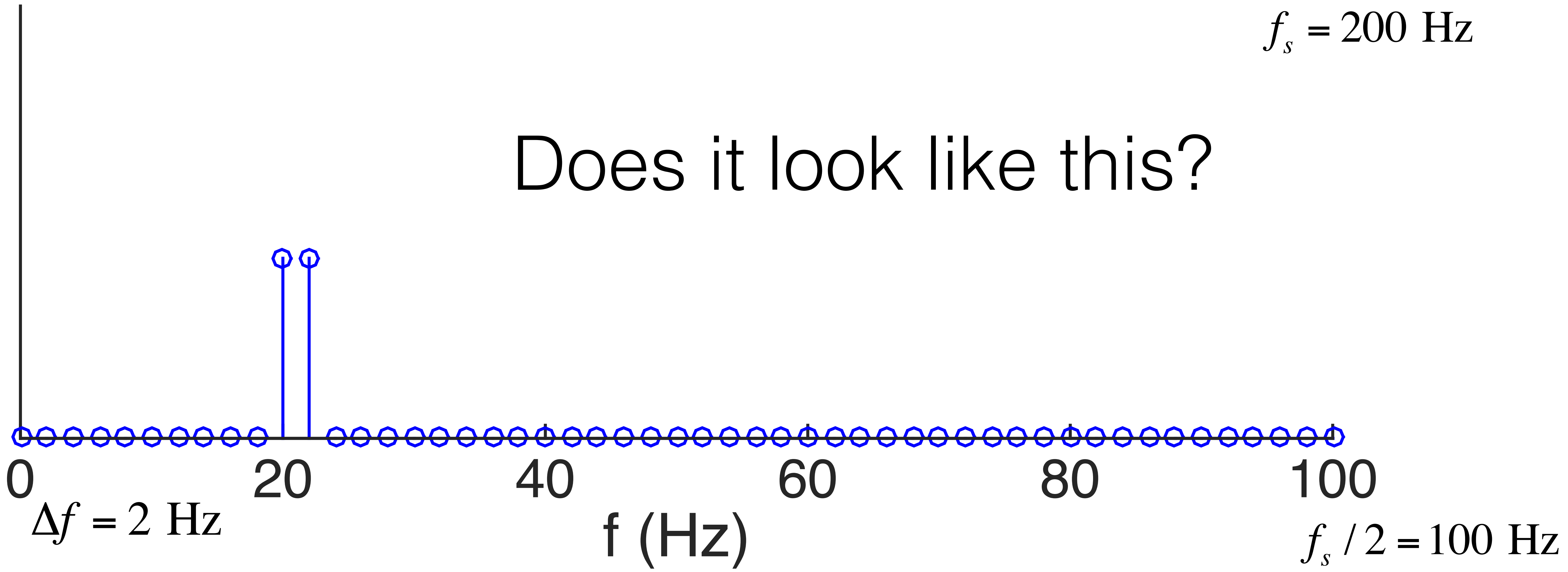
Example 2

A pure sinusoid (single frequency).

What does it look like in the Fourier Domain?

$$x[t] = \cos(2\pi f_b t)$$
$$f_b = 21 \text{ Hz}$$
$$\Delta f = 2 \text{ Hz}$$





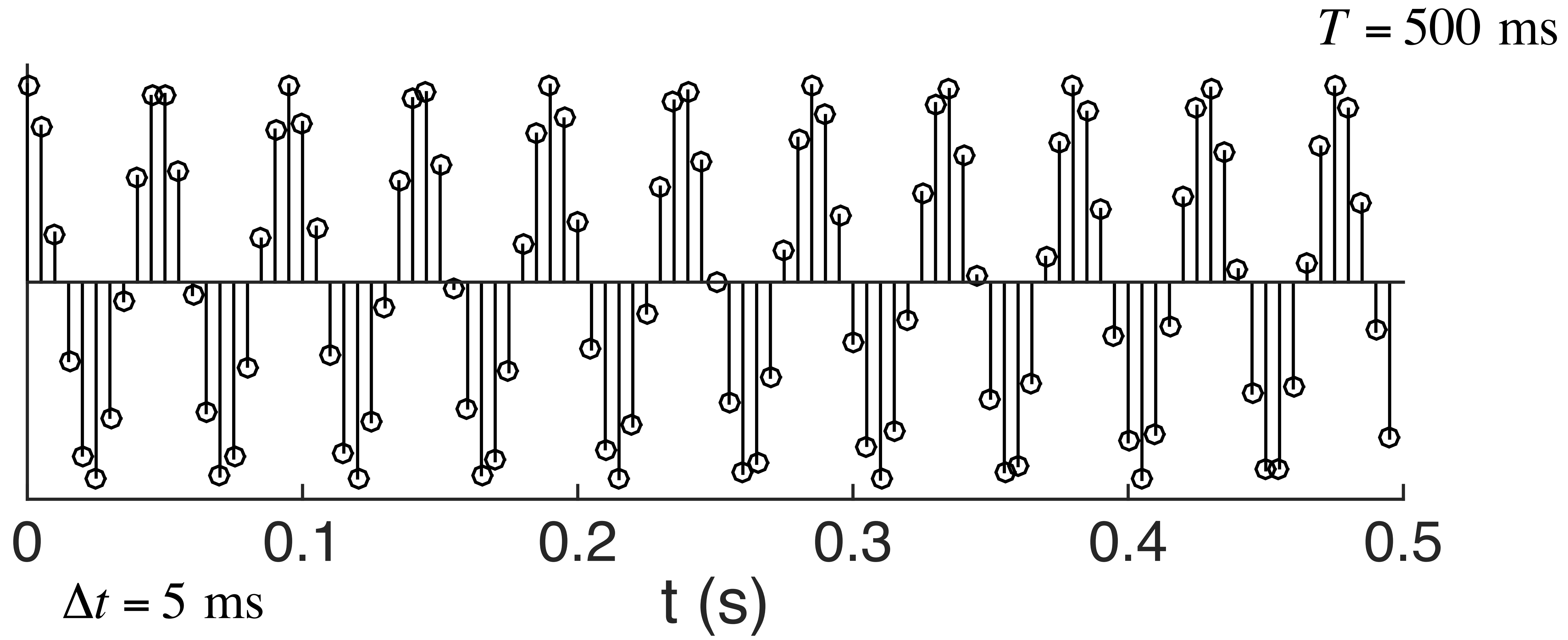
Spectral Leakage

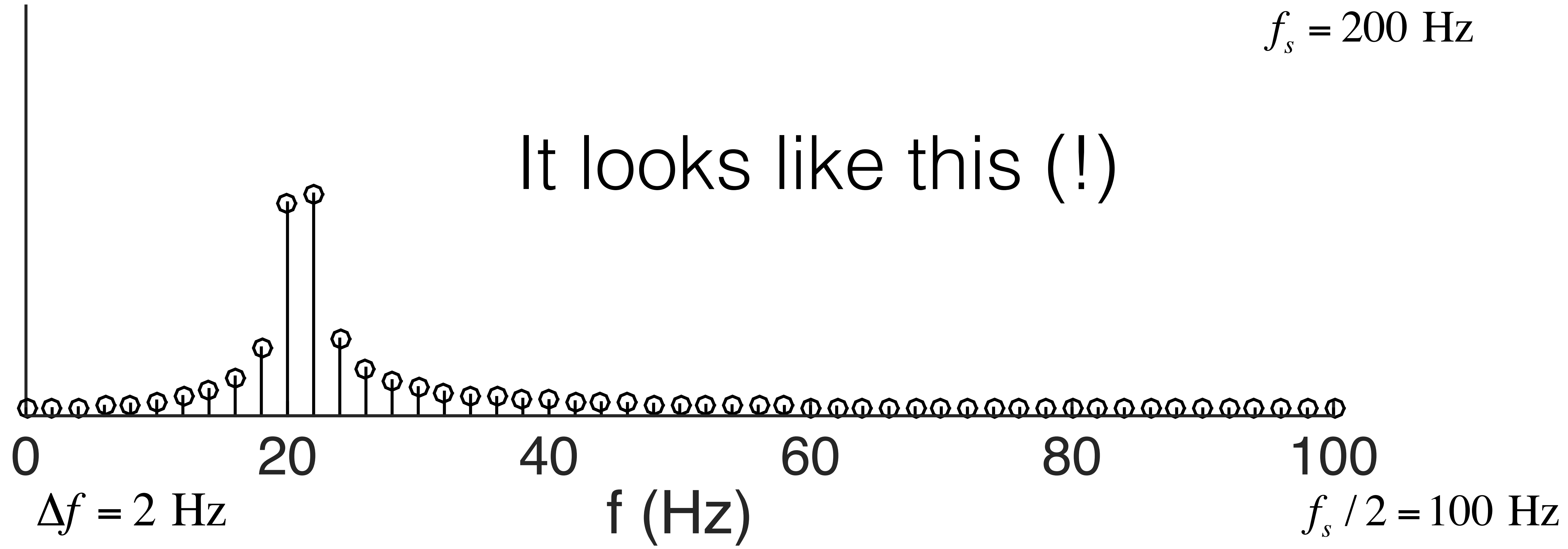
Example 2

A pure sinusoid (single frequency).

What does it look like in the Fourier Domain?

$$x[t] = \cos(2\pi f_b t)$$
$$f_b = 21 \text{ Hz}$$
$$\Delta f = 2 \text{ Hz}$$





Spectral Leakage

Example 2

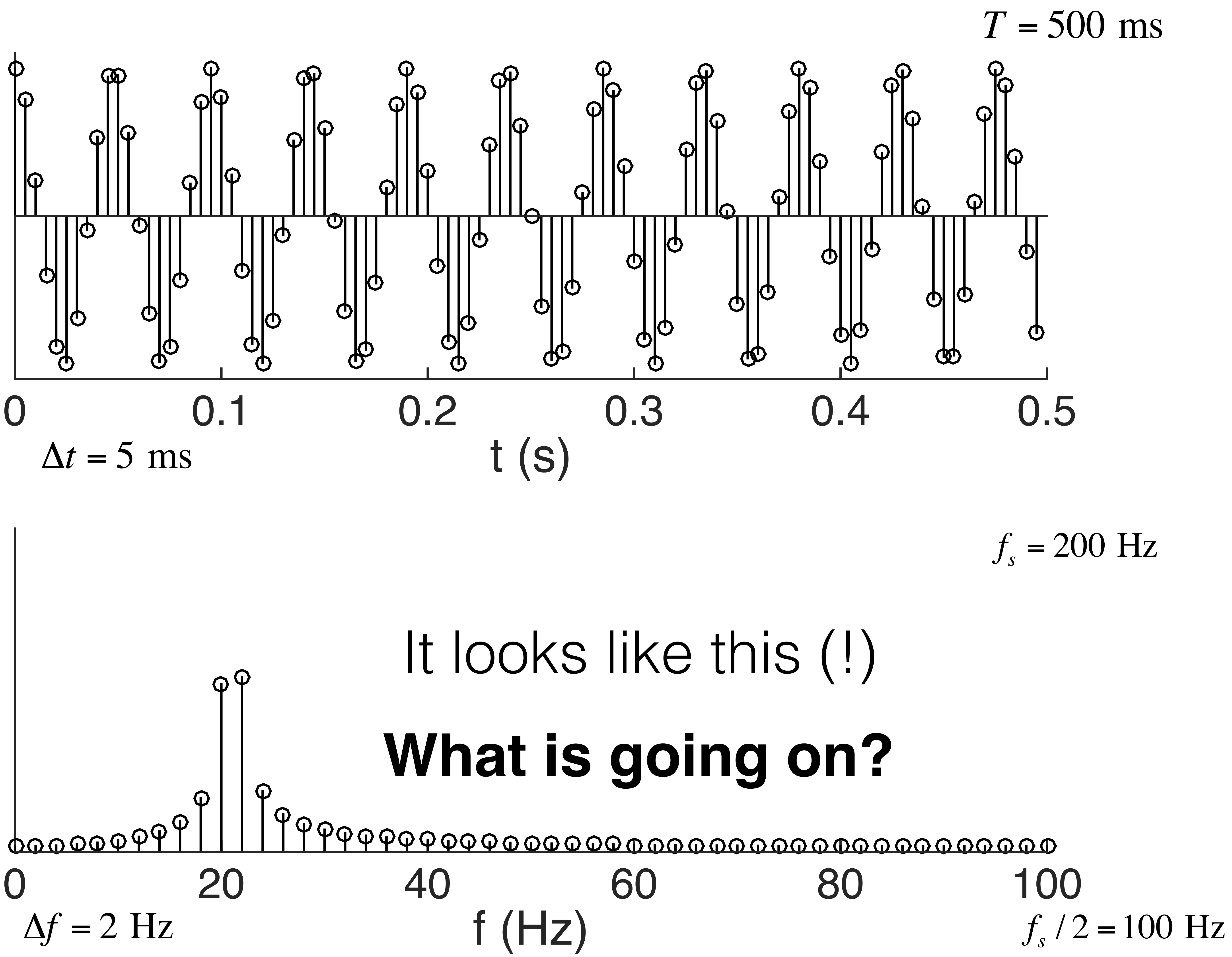
A pure sinusoid (single frequency).

What does it look like in the Fourier Domain?

$$x[t] = \cos(2\pi f_b t)$$

$$f_b = 21 \text{ Hz}$$

$$\Delta f = 2 \text{ Hz}$$



Spectral Leakage

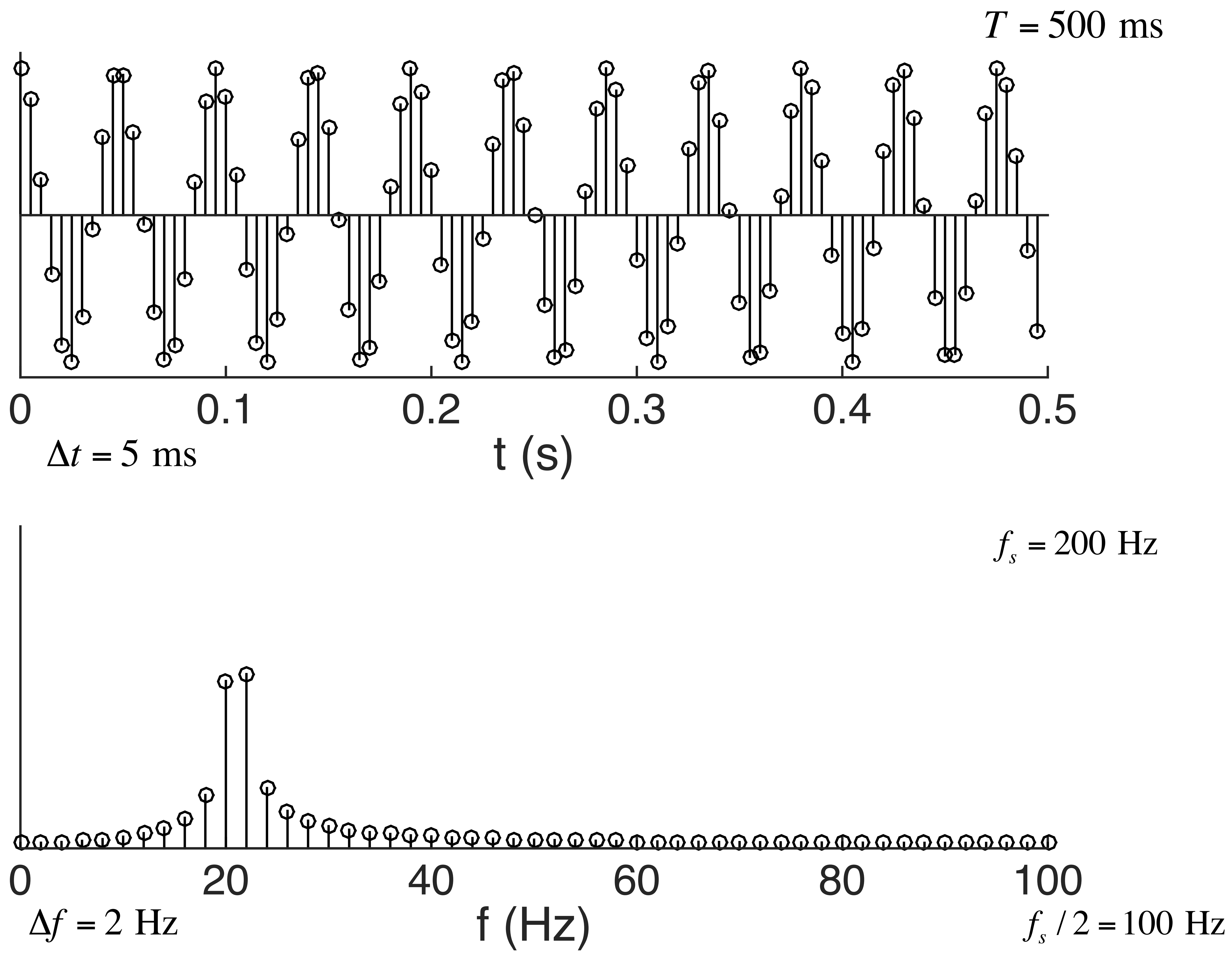
A sinusoid whose single frequency is *not* a Fourier frequency exhibits *Spectral Leakage*.

Spectral Leakage of a strong signal component can easily overwhelm weaker nearby signal components.

$$x[t] = \cos(2\pi f_b t)$$

$$f_b = 21\ \text{Hz}$$

$$\Delta f = 2\ \text{Hz}$$



Spectral Leakage

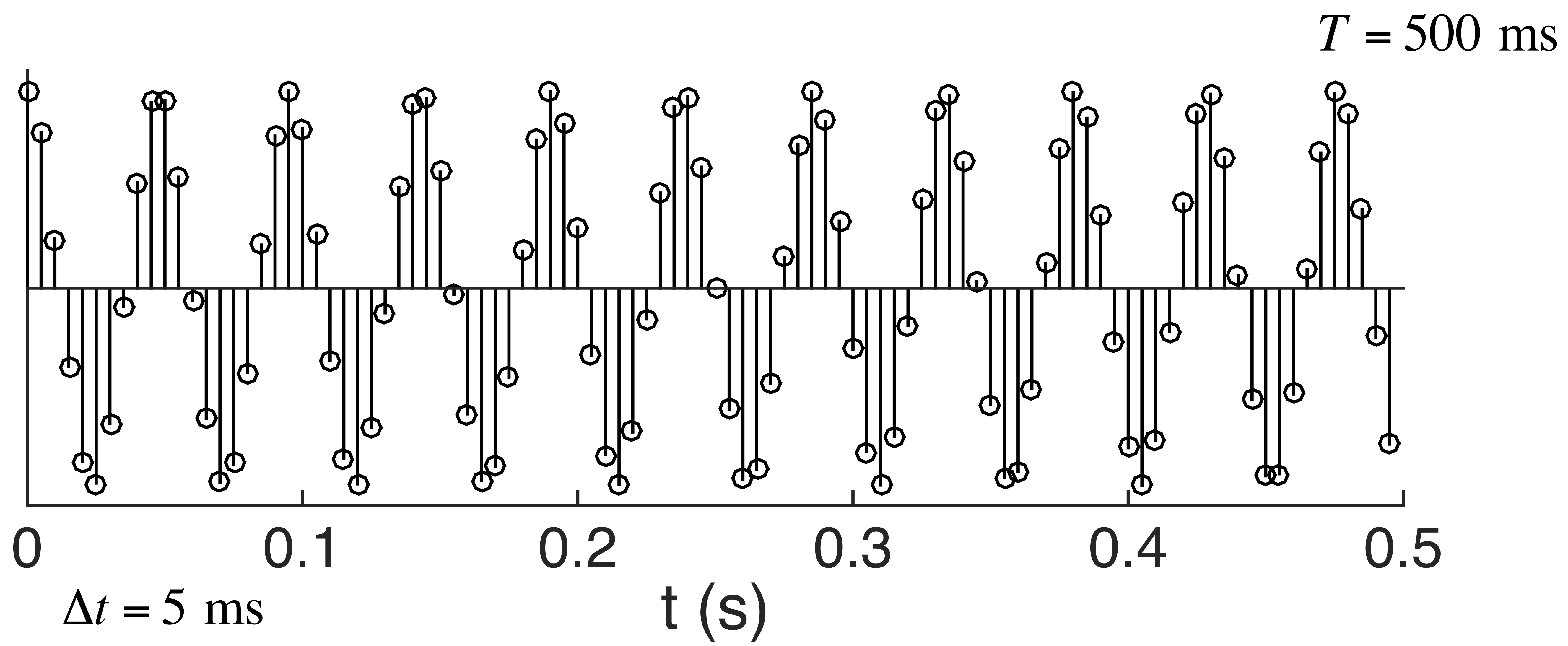
What is the origin of spectral leakage?

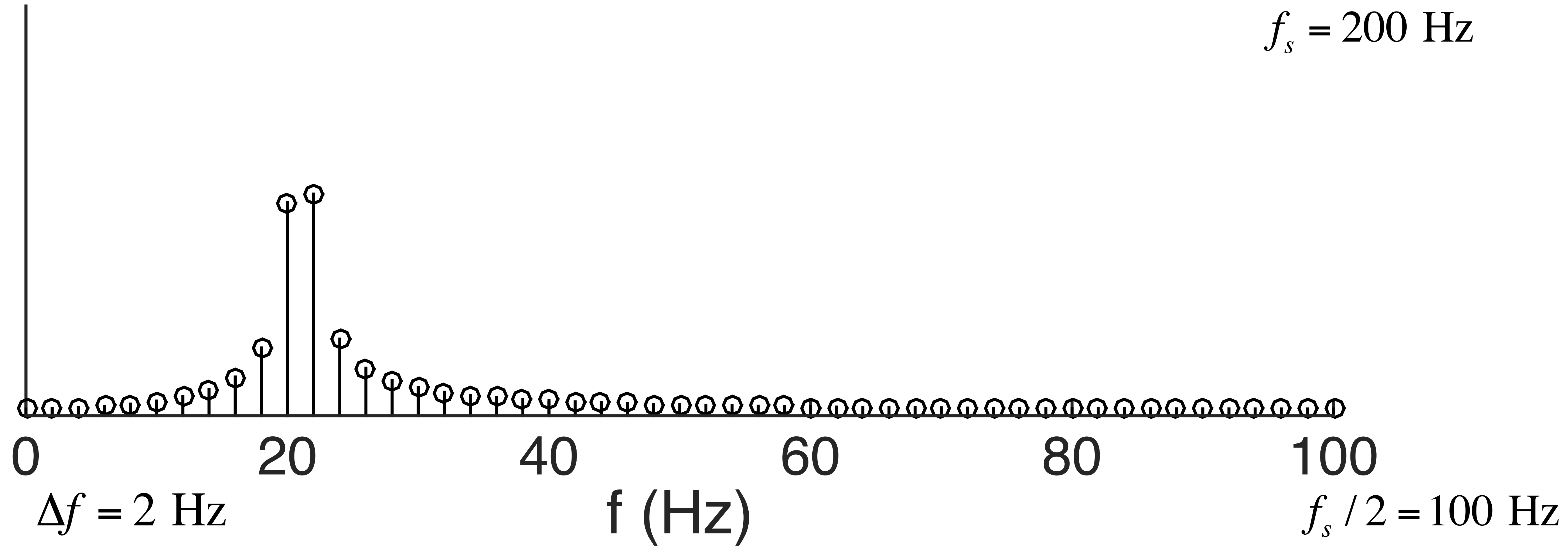
This signal is a cosine, but not periodic with period 2π. The ends do not match.

This can be seen by rotating the signal by *T*/2, which does affect the Fourier transform in magnitude.

Signal discontinuities are spectrally broadband!

$$f_b = 21 \text{ Hz}$$
$$\Delta f = 2 \text{ Hz}$$





Spectral Leakage

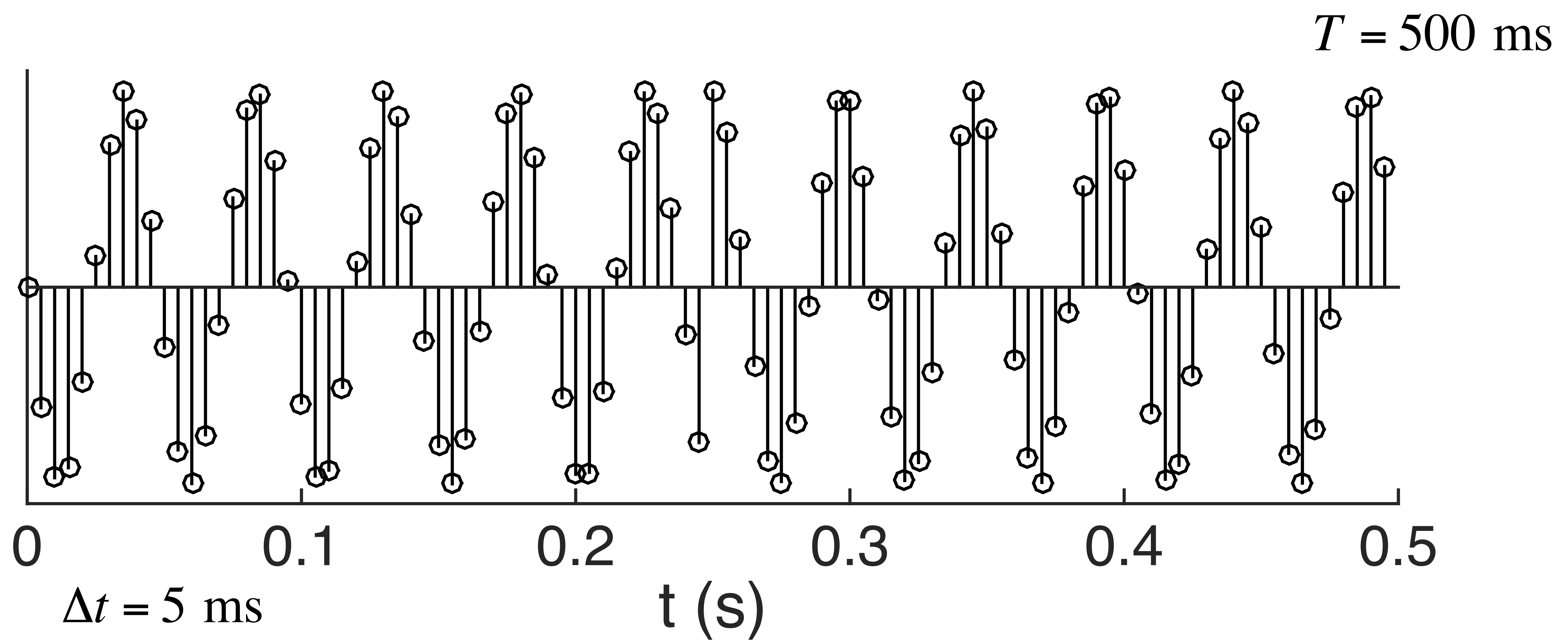
What is the origin of spectral leakage?

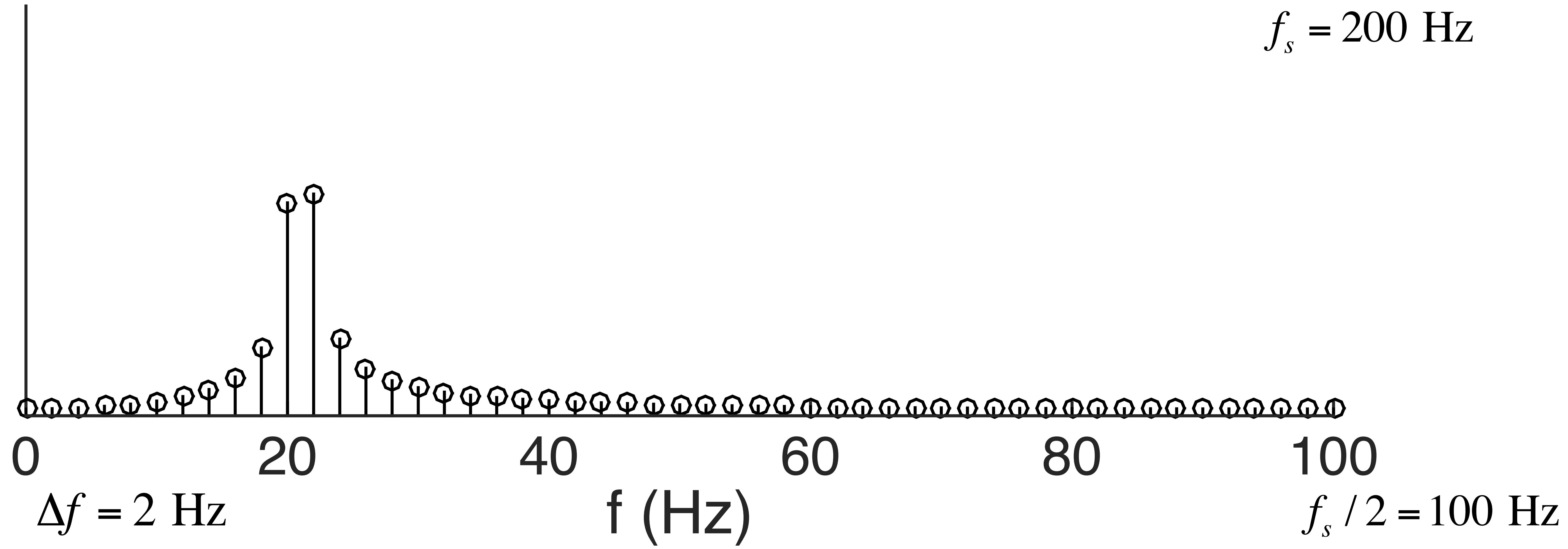
This signal is a cosine, but not periodic with period 2π. The ends do not match.

This can be seen by rotating the signal by *T*/2, which does affect the Fourier transform in magnitude.

Signal discontinuities are spectrally broadband!

$$f_b = 21 \text{ Hz}$$
$$\Delta f = 2 \text{ Hz}$$





Spectral Leakage

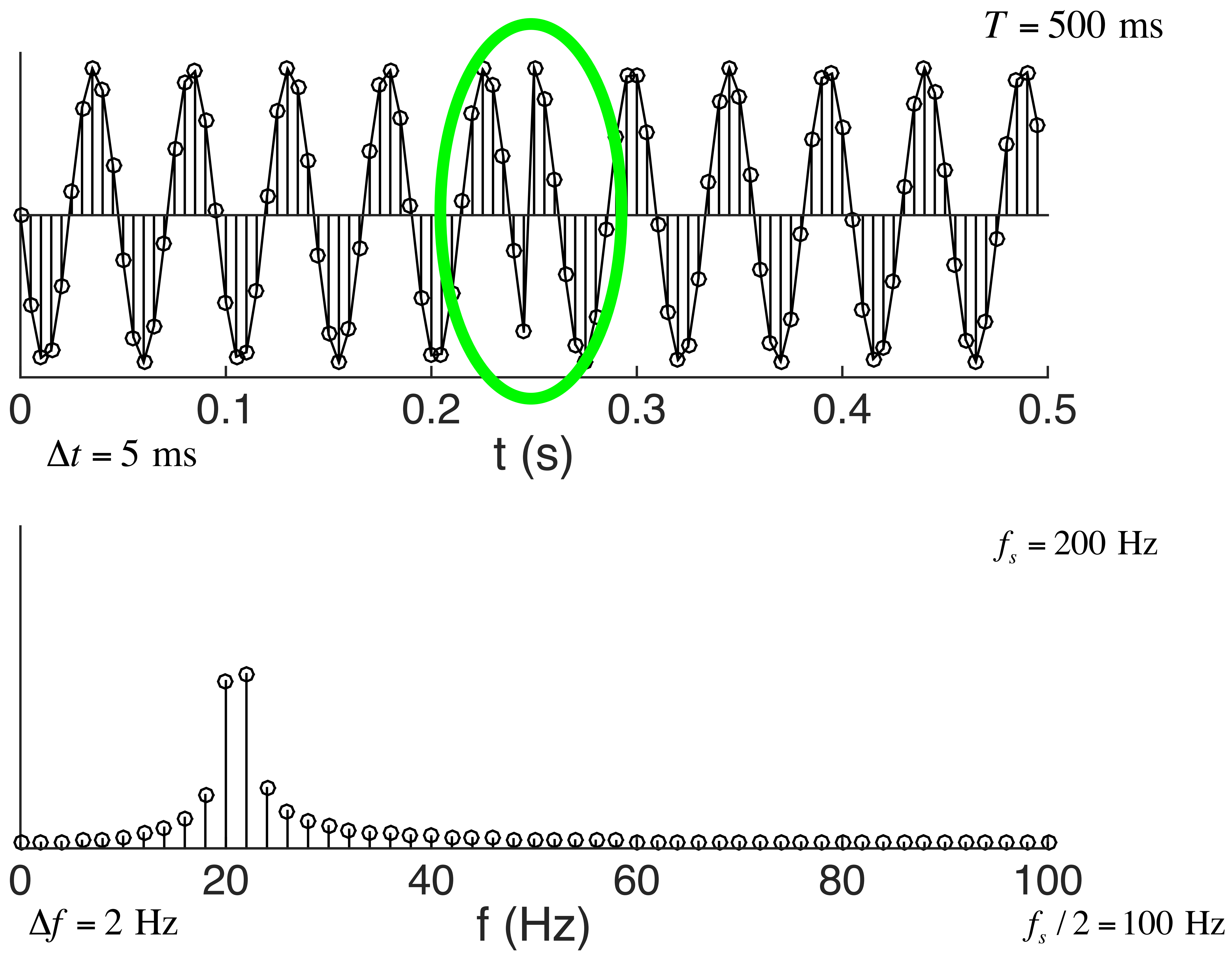
What is the origin of spectral leakage?

This signal is a cosine, but not periodic with period 2π. The ends do not match.

This can be seen by rotating the signal by *T*/2, which does affect the Fourier transform in magnitude.

Signal discontinuities are spectrally broadband!

$$f_b = 21 \text{ Hz}$$
$$\Delta f = 2 \text{ Hz}$$

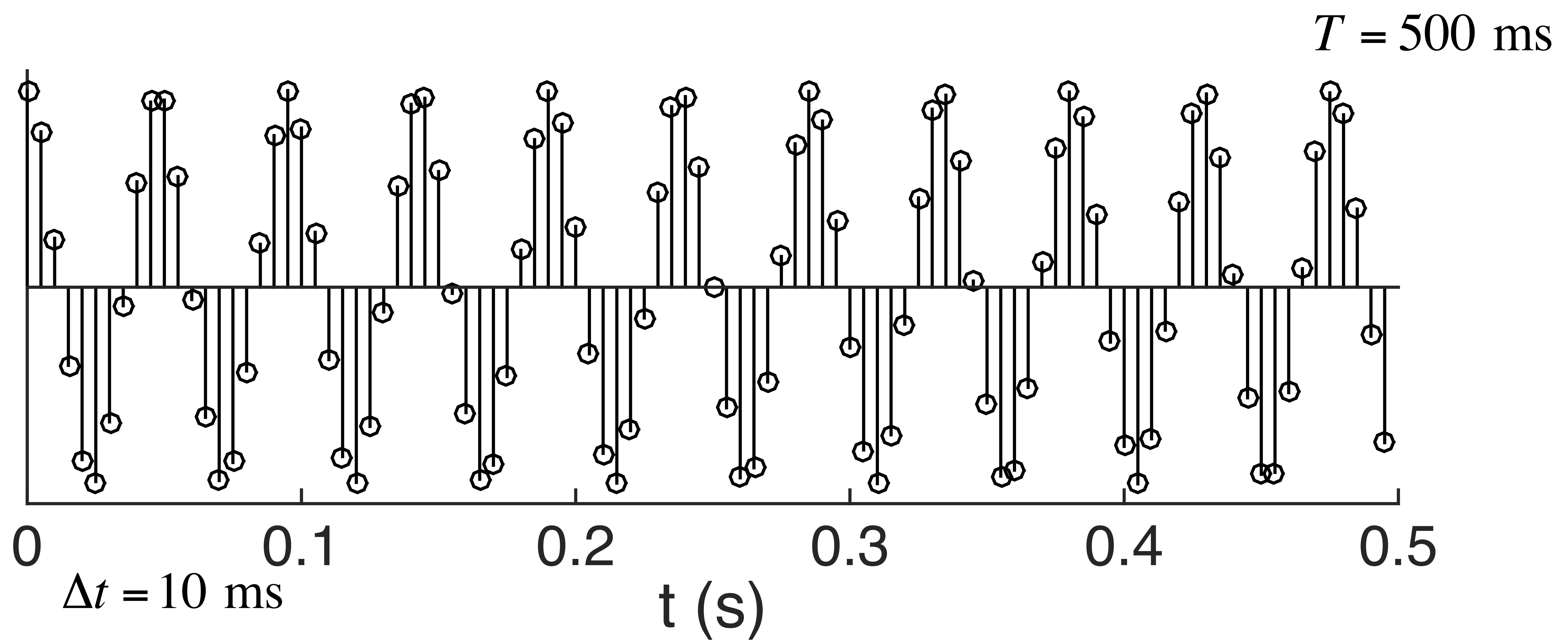


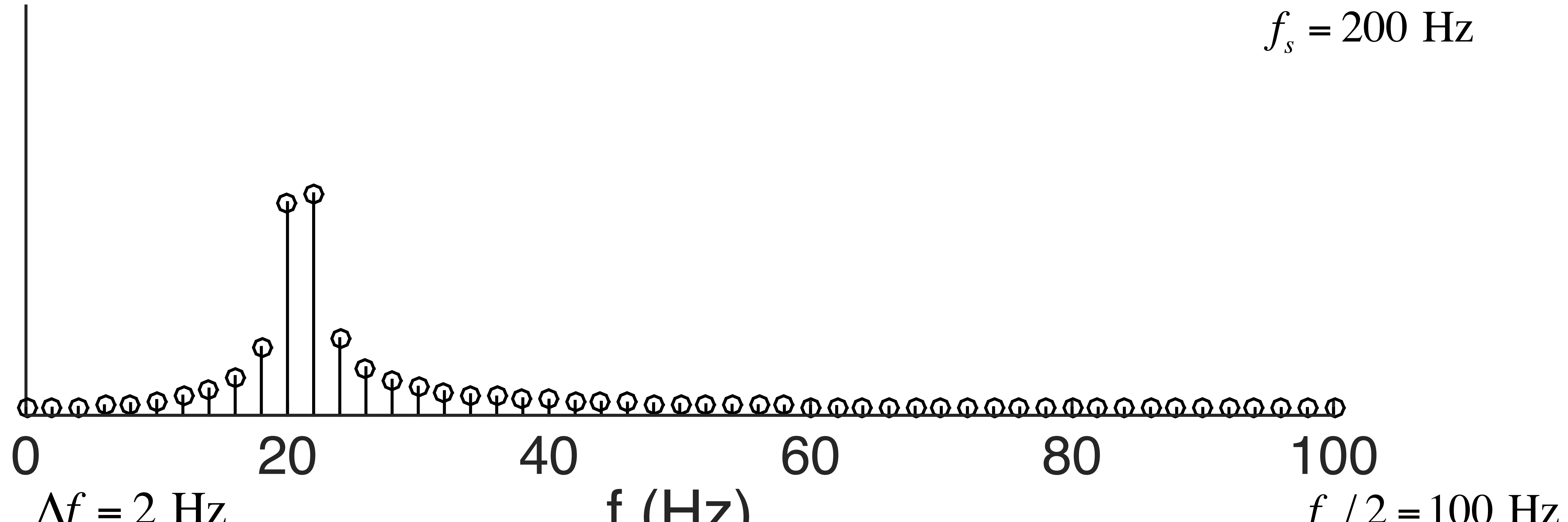
Spectral Leakage

How do we ameliorate the edge “discontinuity”?

Modulate the signal by a window (i.e.,“window” the signal).

$$x[t] = \cos(2\pi f_b t)$$
$$f_b = 21 \text{ Hz}$$
$$\Delta f = 2 \text{ Hz}$$

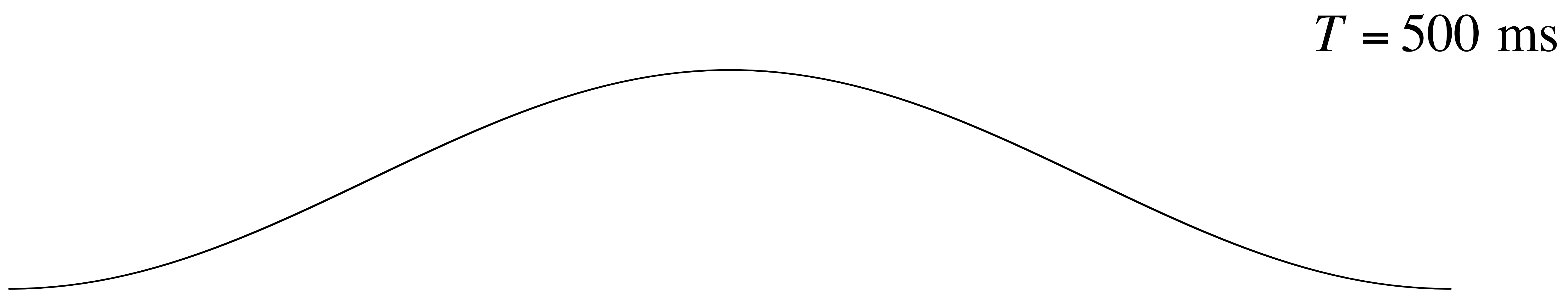




Spectral Leakage

How do we ameliorate the edge "discontinuity"?

Modulate the signal by a window (i.e.,"window" the signal).



$$x[t] = \cos(2\pi f_b t)$$

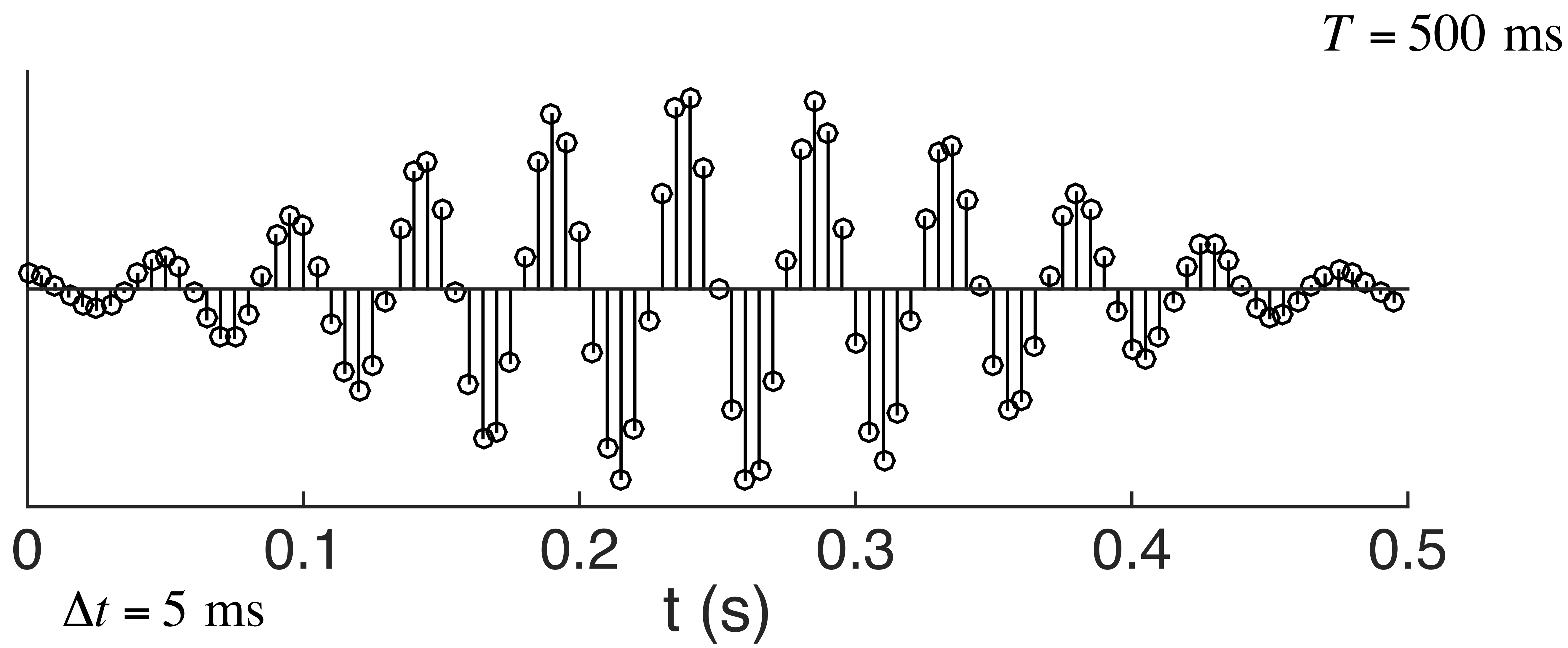
$$f_b = 21 \text{ Hz}$$

$$\Delta f = 2 \text{ Hz}$$

Spectral Leakage

How do we ameliorate the edge “discontinuity”?

Modulate the signal by a window (“window” the signal).



$$x[t] = \cos(2\pi f_b t)$$

$$f_b = 21 \text{ Hz}$$

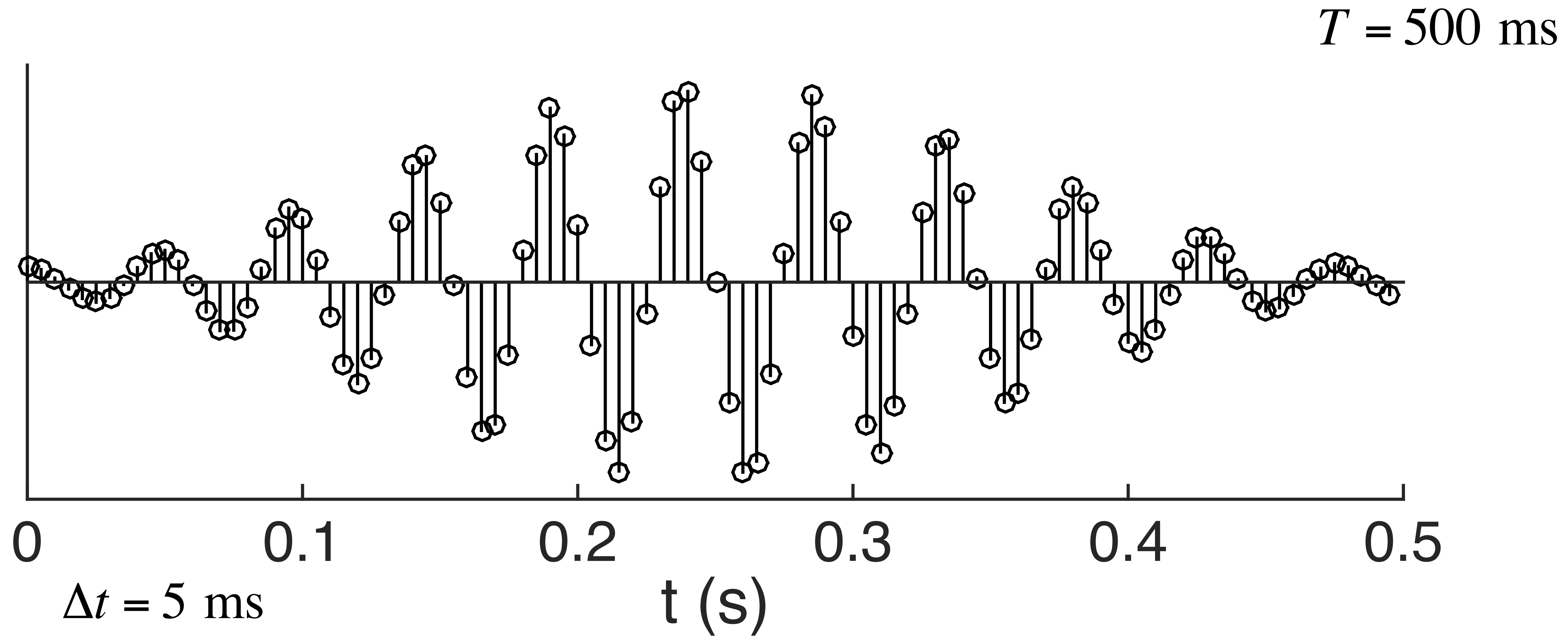
$$\Delta f = 2 \text{ Hz}$$

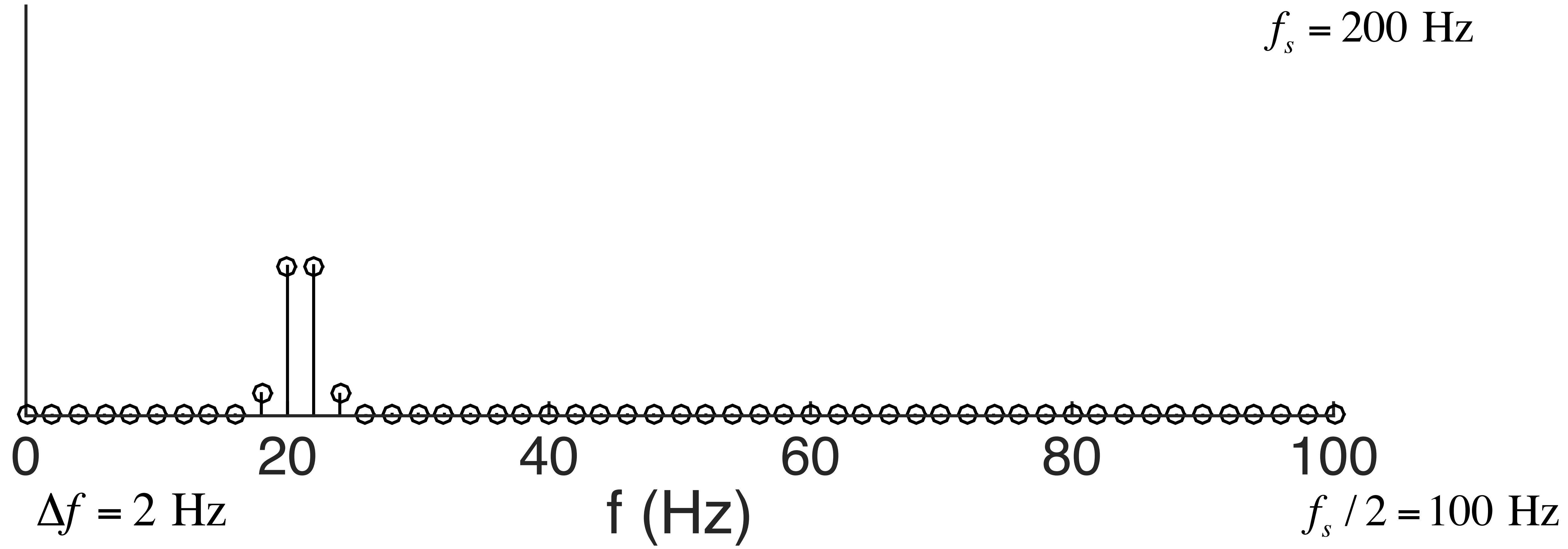
Spectral Leakage

How do we ameliorate the edge "discontinuity"?

Modulate the signal by a window ("window" the signal).

$$x[t] = \cos(2\pi f_b t)$$
$$f_b = 21 \text{ Hz}$$
$$\Delta f = 2 \text{ Hz}$$





Spectral Leakage

How do we ameliorate the edge “discontinuity”?

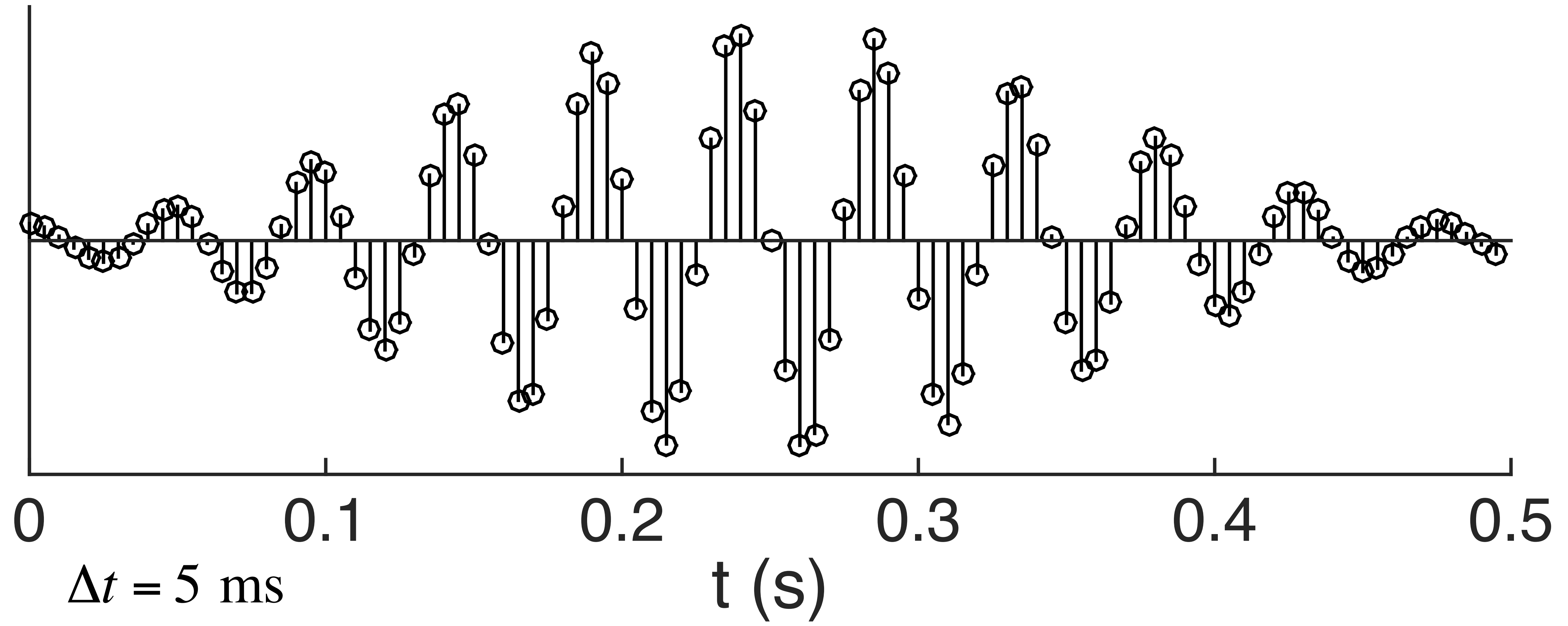
Modulate the signal by a window (“window” the signal).

$$x[t] = \cos(2\pi f_b t)$$

$$f_b = 21 \text{ Hz}$$

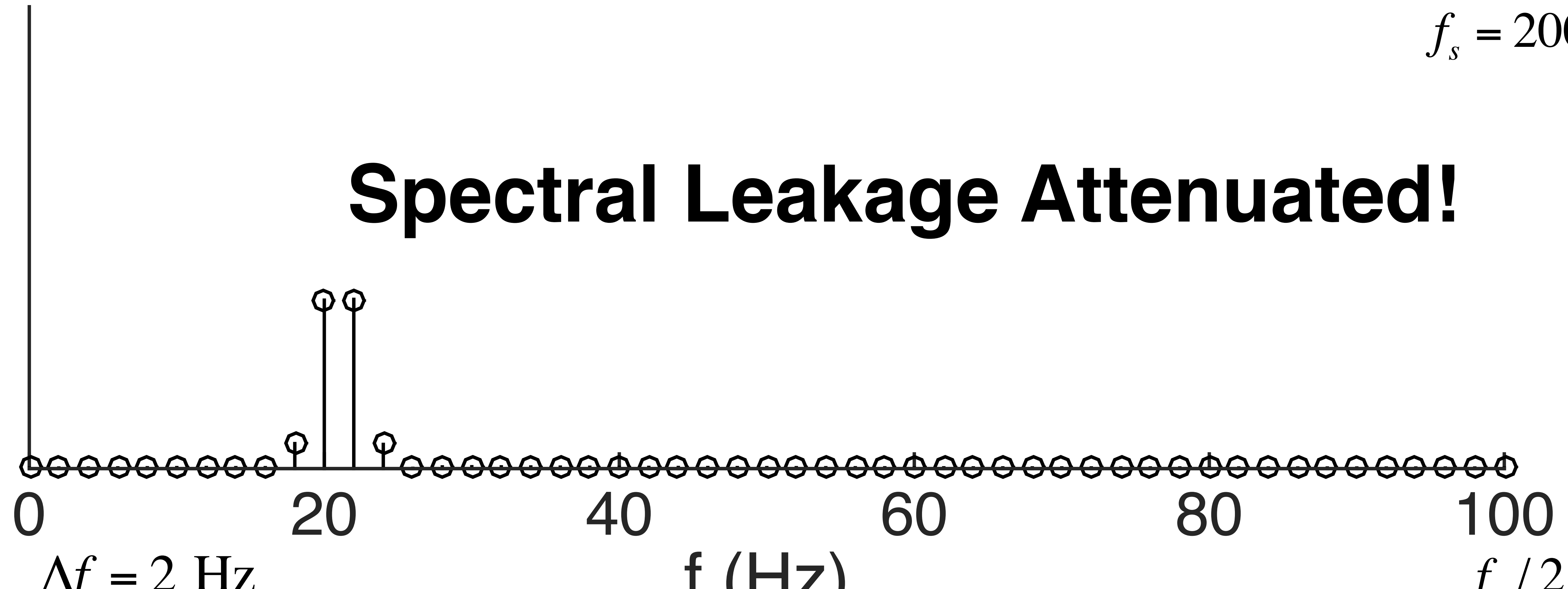
$$\Delta f = 2 \text{ Hz}$$

$T = 500$ ms



$\Delta t = 5$ ms

$f_s = 200$ Hz



$\Delta f = 2$ Hz

$f_s / 2 = 100$ Hz

Windowing & Frequency Resolution

- Windowing to attenuate spectral leakage is critical for frequency estimation (spectral power, spectrogram, etc.).
- Achieved by *blurring neighboring frequencies*/decreasing *effective* frequency resolution (typically by ~2x).
- If you ultimately need an final spectral resolution of Δf, you actually require a signal duration of ~$2/\Delta f$ (not just $1/\Delta f$).
- For example, 1 Hz resolution, without spectral leakage corruption, requires ~2 s signal duration. 2 Hz resolution, without spectral leakage corruption, requires ~4 s signal duration.

Outline

- Fourier Transform: *Why It's Useful*, and *What it Can/Cannot Do For You*
- Filters: *What They Do*, and *How They Do It*
- Filters: *Why So Many Different Kinds?   Which Should I Use and When*?
- Grab Bag:
    - *Use Causal Filters; Windowing is Good*; *Low-Pass your Envelopes*

Conclusions

- Fourier Transforms and Filtering is Complicated
- But not Too Complicated
- Mathematical Definitions will always Win/Tie over Intuition
- But Guided Intuition will put on a Strong Show
- Debugging using Guided Intuition faster than using Math