Research Background

- MEG-based Auditory Neuroscience
- Cocktail-Party Auditory Processing
- Role of Attention
- Neural Representations of Speech
- Fundamentally Temporally Neural Representations

More at <http://www.isr.umd.edu/Labs/CSSL/simonlab/>
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*
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*
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 (\(\Delta t\)) & frequency span (\(f_s\))

• Reciprocal relationship: frequency resolution (\(\Delta f\)) & time span (\(T\))

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

where:

\[
t = 0, \Delta t, 2\Delta t, \ldots, T - \Delta t
\]

\[
f_k = 0, \Delta f, 2\Delta f, \ldots, f_s - \Delta f
\]

\[
f_s = \text{sampling frequency} = \frac{1}{\Delta t}
\]

\[
T = \text{signal duration} = \frac{1}{\Delta f}
\]
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 ($\Delta t$) & frequency span ($f_s$)
- Reciprocal relationship: frequency resolution ($\Delta f$) & time span ($T$)

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

where:

- \( t = 0, \Delta t, 2\Delta t, \ldots, T - \Delta t \)
- \( f_k = 0, \Delta f, 2\Delta f, \ldots, f_s - \Delta f \)
- \( f_s = \text{sampling frequency} = \frac{1}{\Delta t} \)
- \( T = \text{signal duration} = \frac{1}{\Delta f} \)
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 ($\Delta t$) & *sample frequency* ($f_s$)
- Reciprocal relationship: *frequency resolution* ($\Delta f$) & *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 ($\Delta t$) & sample frequency ($f_s$)
- Reciprocal relationship: frequency resolution ($\Delta f$) & 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 ($\Delta t$) & *sample frequency* ($f_s$)
- Reciprocal relationship: *frequency resolution* ($\Delta f$) & *duration* ($T$)

\[ f_s = 100 \text{ Hz} \]

\[ \Delta t = 10 \text{ ms} \]

\[ T = 500 \text{ ms} \]

\[ f_s / 2 = 50 \text{ Hz} \]
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 ($\Delta t$) & *sample frequency* ($f_s$)
- Reciprocal relationship: *frequency resolution* ($\Delta f$) & *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 ($\Delta t$) & *sample frequency* ($f_s$)

• Reciprocal relationship: *frequency resolution* ($\Delta f$) & *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 ($\Delta t$) & *sample frequency* ($f_s$)

• Reciprocal relationship: *frequency resolution* ($\Delta f$) & *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 ($\Delta t$) & sample frequency ($f_s$)
- Reciprocal relationship: frequency resolution ($\Delta f$) & duration ($T$)
Fourier Transform: Time-Frequency Tradeoff

$\Delta t = 10 \text{ ms}$  \quad  $T = 500 \text{ ms}$

$f_s = 100 \text{ Hz}$  \quad  $\Delta f = 2 \text{ Hz}$

$f_s / 2 = 50 \text{ Hz}$

$f_s = 100 \text{ Hz}$
Fourier Transform: Time-Frequency Tradeoff

\[ T = 100 \text{ ms} \quad \Delta t = 10 \text{ ms} \]

\[ f_s = 100 \text{ Hz} \quad \Delta f = 2 \text{ Hz} \]

\[ f_s / 2 = 50 \text{ Hz} \]

\[ f_s = 100 \text{ Hz} \quad T = 1000 \text{ ms} \]

\[ f_s / 2 = 50 \text{ Hz} \]
Fourier Transform: Time-Frequency Tradeoff

- $\Delta t = 10 \text{ ms}$, $T = 500 \text{ ms}$
- $f_s = 100 \text{ Hz}$, $\Delta f = 2 \text{ Hz}$
- $f_s / 2 = 50 \text{ Hz}$

- $T = 1000 \text{ ms}$
- $\Delta f = 1 \text{ Hz}$
- $f_s / 2 = 50 \text{ Hz}$
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
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
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*
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
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…
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…
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**
  ![Low Pass Filter Diagram](image)

  **Band Pass**
  ![Band Pass Filter Diagram](image)

  **High Pass**
  ![High Pass Filter Diagram](image)

  **Band Stop**
  ![Band Stop Filter Diagram](image)

  **Notch**
  ![Notch Filter Diagram](image)

  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
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

Soft Transition
Filters: How Selective?

• How sharp a transition?

  “Ideal” Filter

  ![Ideal Filter Plot]

  Sharp Transition

  ![Sharp Transition Plot]

  Soft Transition

  ![Soft Transition Plot]
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

\[ y[t] = \frac{1}{2} x[t] + \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Average

\[ y[t] = \frac{1}{2} x[t] + \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Average

\[ y[t] = \frac{1}{2} x[t] + \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Average

\[ y[t] = \frac{1}{2} x[t] + \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Average

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

Diagram of \( x[t] \) and \( x[t - \Delta t] \) showing moving averages.
Example: Two-Point Moving Average

\[ y[t] = \frac{1}{2} x[t] + \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Average

\[ x[t] \]

\[ x[t - \Delta t] \]
Example: Two-Point Moving Average

$$x[t]$$

$$x[t]$$

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

$$x[t - \Delta t]$$
Example: Two-Point Moving Average

\[
x[t] = \frac{x[t]}{2} + \frac{x[t - \Delta t]}{2}
\]
Example: Two-Point Moving Average

\[ x[t] = \frac{x[t]}{2} + \frac{x[t - \Delta t]}{2} \]
Example: Two-Point Moving Average

\[ y[t] = \frac{1}{2} x[t] + \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Average

\[ y[t] = \frac{1}{2} x[t] + \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Average

\[ y[t] = \frac{1}{2} x[t] + \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Average

\[ y[t] = \frac{1}{2} x[t] + \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Average

\[ y[t] = \frac{1}{2} x[t] + \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Average

\[y[t] = \frac{1}{2} x[t] + \frac{1}{2} x[t - \Delta t]\]
Example: Two-Point Moving Average

\[ y[t] = \frac{1}{2} x[t] + \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Average

\[ y[t] = \frac{1}{2} x[t] + \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Average

\[ y[t] = \frac{1}{2} x[t] + \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Average

\[ y[t] = \frac{1}{2} x[t] + \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Average

\[ y[t] = \frac{1}{2} x[t] + \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Average

\[ y[t] = \frac{1}{2} x[t] + \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Average

\[ y[t] = \frac{1}{2} x[t] + \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Average

\[ y[t] = \frac{1}{2} x[t] + \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Average

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

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

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

Results:

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

Low Pass Filter?
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* ($\Delta t$) & *frequency span* ($f_s$)

• Reciprocal relationship: *frequency resolution* ($\Delta f$) & *time span* ($T$)
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 ($\Delta t$) & *frequency* span ($f_s$)
- Reciprocal relationship: *frequency* resolution ($\Delta f$) & *time* span ($T$)

\[
\Delta t = 10 \text{ ms} \quad T = 160 \text{ ms} \quad f_s = 100 \text{ Hz}
\]
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 ($\Delta t$) & *frequency* span ($f_s$)
- Reciprocal relationship: *frequency* resolution ($\Delta f$) & *time* span ($T$)

\[ T = \frac{1}{f_s} = 160 \text{ ms} \]
\[ \Delta t = 10 \text{ ms} \]
\[ f_s = 100 \text{ Hz} \]
\[ \frac{f_s}{2} = 50 \text{ Hz} \]
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 ($\Delta t$) & *frequency span* ($f_s$)
- Reciprocal relationship: *frequency resolution* ($\Delta f$) & *time span* ($T$)

\[
\Delta t = 10 \text{ ms} \quad T = 160 \text{ ms} \\
\Delta f = 6.25 \text{ Hz} \\
f_s = 100 \text{ Hz} \\
f_s / 2 = 50 \text{ Hz}
\]
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 ($\Delta t$) & *frequency* span ($f_s$)

• Reciprocal relationship: *frequency* resolution ($\Delta f$) & *time* span ($T$)

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

where:

\[
t = 0, \Delta t, 2\Delta t, \ldots, T - \Delta t
\]

\[
f_k = 0, \Delta f, 2\Delta f, \ldots, f_s - \Delta f
\]

\[f_s = \text{sampling frequency} = \frac{1}{\Delta t}\]

\[T = \text{signal duration} = \frac{1}{\Delta f}\]
The Fourier Transform
The Fourier Transform

Every Signal = Fourier Coefficients
Filters and the Fourier Transform

\[ \text{Filter}(\text{signal}) = \text{Filter}( \text{Fourier Coefficients} ) \times \text{Fourier Coefficients} \]
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
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

\[ y[t] = \frac{1}{2} x[t] - \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Difference

\[ x[t] \]

\[ x[t - \Delta t] \]
Example: Two-Point Moving Difference

\[ x[t] \]

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

\[ x[t] \]

\[ x[t - \Delta t] \]
Example: Two-Point Moving Difference

\[x[t] - x[t - \Delta t]/2\]
Example: Two-Point Moving Difference

\[ x[t] \]

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

\[ \frac{x[t]}{2} - \frac{x[t - \Delta t]}{2} \]
Example: Two-Point Moving Difference

\[ y[t] = \frac{1}{2} x[t] - \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Difference

\[ y[t] = \frac{1}{2} x[t] - \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Difference

\[ y[t] = \frac{1}{2} x[t] - \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Difference

\[ y[t] = \frac{1}{2} x[t] - \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Difference

\[ y[t] = \frac{1}{2} x[t] - \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Difference

\[ y[t] = \frac{1}{2} x[t] - \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Difference

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

\[ y[t] = \frac{1}{2} x[t] - \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Difference

\[ y[t] = \frac{1}{2} x[t] - \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Difference

\[ y[t] = \frac{1}{2} x[t] - \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Difference

\[ y[t] = \frac{1}{2} x[t] - \frac{1}{2} x[t - \Delta t] \]
Example: Two-Point Moving Difference

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

**Results:**
- Exaggerates differences
- Amplifies quickly varying signals
- Attenuates slowly varying signals
- High Pass Filter?
Example: Two-Point Moving Difference

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

Results:
- Exaggerates differences
- Amplifies quickly varying signals
- Attenuates slowly varying signals
- High Pass Filter?
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
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
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
“Which Filter Should I Use?”

—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 a valid tradeoff

  • Don’t go overboard one way or the other

• Some filter decisions allow one 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

Neural Response
Ringing Artifacts

Neural Response

Measured Neural Response
Ringing Artifacts

Measured Neural Response
Ringing Artifacts

Measured Neural Response

Response Spectrum
Ringing Artifacts

- Measured Neural Response
- Filter with Sharp Frequency Transition
- Response Spectrum
- Filter with Soft Frequency Transition
Ringing Artifacts

Measured Neural Response

Filter with Sharp Frequency Transition

Response Spectrum

Filter with Soft Frequency Transition
Ringing Artifacts

Measured Neural Response

Filter with Sharp Frequency Transition

Response Spectrum

Filter with Soft Frequency Transition
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)

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): group delay value and group delay frequency dependence

  • signal loss due to filter startup (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

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

\[ \tau_D: \frac{\Delta t}{2} (?) \]
Group Delay Examples

\[ y[t] = \frac{1}{2} x[t] + \frac{1}{2} x[t - \Delta t] \]
Group Delay Examples

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

\[ \tau_D : \Delta t \]

\[ \tau_D : \Delta t \]
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, \( N_{\text{order}} \), is the number of time shifts used by the most delayed component, minus 1 (i.e. one less than the length of the filter).
- The group delay of an FIR filter is \( \Delta t \times \frac{N_{\text{order}}}{2} \).
  - The higher the order, the longer the group delay
  - Calculating latencies? You may need to compensate (OK for *peak* latencies).
  - The smaller \( \Delta t \), the smaller the delay, so if possible filter at high sampling frequency.
Group Delay: FIR filters

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

\[ y[t] = \frac{1}{2} x[t] + \frac{1}{2} x[t - \Delta t] \]
Group Delay: FIR filters

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

\[ y[t] = \frac{1}{2} x[t] + \frac{1}{2} x[t - \Delta t] \]
Group Delay: FIR filters

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

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

- The group delay of an IIR filter may be irrelevant over frequencies that are “stopped”.

[Graph showing group delay and magnitude of an IIR filter]
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 group delay of an IIR filter is longest during the frequency transition.
Group Delay: IIR filters

- The group delay of an IIR filter is longest during the frequency transition.
- The sharper the transition, the longer the group delay
Group Delay: IIR filters

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

![Graphs showing group delay and magnitude for different frequency transitions.](image)
Group Delay: IIR filters

- The group delay of an IIR filter is longest during the frequency transition.
- The sharper the transition, the longer the group delay
Group Delay: IIR filters

• The group delay of an IIR filter is longest during the frequency transition.

• The sharper the transition, the longer the group delay

• Calculating latencies? You may need to compensate (still possible for peaks dominated by frequencies far from the transition).
Group Delay: IIR filters

• The group delay of an IIR filter is longest during the frequency transition.

• The sharper the transition, the longer the group delay.

• Calculating latencies? You may need to compensate (still possible for peaks dominated by frequencies far from the transition).

• The group delay of an IIR filter does not linearly scale with $\Delta t(!)$, so no penalty for filtering at low sampling frequency.
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 $N_{\text{order}} \times \Delta t$.

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

- Recommendation: either keep extra earlier data of duration $N_{\text{order}} \times \Delta t$, 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 $N_{\text{order}}$.

- 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

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

- 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), *however 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.

- Even data from the *end* of the signal may help substantially 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)
How Would I even Notice Instability?

It’s not subtle (but only if you know where to look)
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 the case 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 the case 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.
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% let 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% let through) do you *really* need? If your noise is suppressed only by 100x, would you even 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; Low-Pass your Envelopes
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
Causal & non-Causal Filtering

All filters discussed here are causal.

- Changes to the input signal cause changes in the output. The output changes occur at the same moment, or later, but never earlier.

- Some output changes are desirable: using a low pass filter to slow down fast changes in the input signal.

- Some output changes are undesirable: ringing due to increase of transition-frequency content in the input signal.
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

Zero-Phase Notched FM Sweep

Ringing:
• persistent
• causal

Ringing:
• duplicated and flipped
• no cancellation (except “on average”?)
Causal & non-Causal Filtering

- Zero-phase filters do not remove distortions, but instead spread them out symmetrically.

- Spreading them out gives zero “on average” but not actually zero.

- Other temporal features (e.g. smoothing, change detection) get similarly spread out by duplication.

  - May remove peak delay, but replaces un-delayed rise with an anti-causal rise.

- Recommendation: Don’t use. Causes far more problems than solutions.
Windowing and Frequency Resolution

• Frequency resolution ($\Delta f$), 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 ($\Delta f$) and sampling frequency ($f_s$).
Windowing and Frequency Resolution

Finer frequency resolution is obtained 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 $\Delta t$ remains unchanged.

• Analogously, it is sometimes desirable to “smear” information over frequencies (e.g. to attenuate spectral leakage).
  • The effective frequency resolution is worse, even though $\Delta f$ remains unchanged.

• This frequency smearing is typically accomplished by windowing in the time domain.
“Fourier coefficients do not always mean what you think they mean.”

–The Princess Bride (paraphrased)
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?

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

\[ f_b = 21 \text{ Hz} \]
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?

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

\[ f_b = 21 \text{ Hz} \]

\[ \Delta f = 2 \text{ Hz} \]

Does it look like this?
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?

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

\[ f_b = 21 \text{ Hz} \]

\[ \Delta f = 2 \text{ Hz} \]

It looks like this (!)

What is going on?
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 $2\pi$. 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!

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

$$\Delta f = 2 \text{ Hz}$$
What is the origin of spectral leakage?

This signal is a cosine, but not periodic with period $2\pi$. 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!

\[
\begin{align*}
f_b &= 21 \text{ Hz} \\
\Delta f &= 2 \text{ Hz}
\end{align*}
\]
Spectral Leakage

What is the origin of spectral leakage?

This signal is a cosine, but not periodic with period $2\pi$. 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!

\[
\begin{align*}
f_b &= 21 \text{ Hz} \\
\Delta f &= 2 \text{ Hz}
\end{align*}
\]
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).

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

\[ f_b = 21 \text{ Hz} \]

\[ \Delta f = 2 \text{ Hz} \]
Windowing & Frequency Resolution

- Windowing to attenuate spectral leakage is critical for frequency estimation (spectral power, spectrogram, etc.).

- Achieves this by blurring the frequency resolution (typically by 2x).

- If you require spectral resolution of $\Delta f$, you also require a signal duration of not just $1/\Delta f$, but really $2/\Delta f$.

- 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.
Low Passing of Envelopes

• An envelope is any slow amplitude modulation of a signal

• No single definition of envelope, except that it is slow and positive

• Commonly used definitions

  • Low passed half-wave rectified signal

  • Low passed magnitude of Analytic Signal (“hilbert” in Matlab).

• Note that the low pass filter is not optional

• An envelope is any slow amplitude modulation of a signal.
Low Passing & Envelopes

Raw Signal  Actual Envelope  Low Passed Analytic Magnitude  Analytic Magnitude
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

• Signal Processing 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