Radar Systems Engineering
Lecture 3
Review of Signals, Systems and
Digital Signal Processing

Dr. Robert M. O’Donnell
IEEE New Hampshire Section
Guest Lecturer
Block Diagram of Radar System

Transmitter
- Power Amplifier
- Waveform Generation
- T/R Switch

Propagation Medium
- Target Radar Cross Section

Antenna
- Receiver
- A/D Converter
- Signal Processor Computer
  - Pulse Compression
  - Clutter Rejection (Doppler Filtering)

General Purpose Computer
- Tracking
- Parameter Estimation
- Thresholding
- Detection

Application of Signals and Systems, and Digital Signal Processing Algorithms to the Received Radar Signals Result in Optimum Target Detection
Reasons for Review Lecture

• Signals and systems, and digital signal processing are usually one semester advanced undergraduate courses for electrical engineering majors

• In no way will this 1+ hour lecture to justice to this large amount of material

• The lecture will present an overview of the material from these two courses that will be useful for understanding the overall Radar Systems Engineering course
  – Goal of lecture- Give non EE majors a quick view of material; they may wish to study in more depth to enhance their understanding of this course.

• UC Berkeley has an excellent, free, video Signals and Systems course (ECE 120) online at //webcast.berkeley.edu
  – Given in Spring 2007
Signal Processing

• Signal processing is the manipulation, analysis and interpretation of signals.

• Signal processing includes:
  – Adaptive filtering / thresholding
  – Spectrum analysis
  – Pulse compression
  – Doppler filtering
  – Image enhancement
  – Adaptive antenna beam forming, and
  – A lot of other non-radar stuff (Image processing, speech processing, etc.)

• It involves the collection, storage and transformation of data
  – Analog and digital signal processing
  – A lot of processing “horsepower” is usually required
Outline

• Continuous Signals
  • Sampled Data and Discrete Time Systems
  • Discrete Fourier Transform (DFT)
  • Fast Fourier Transform (FFT)
  • Finite Impulse Response (FIR) Filters
  • Weighting of Filters
Continuous Time Signal

Examples:

\[ x(t) = 100 \sin(\pi t) - 79 \cos(3\pi t) \]
\[ x(t) = 12t - 300 \]
\[ x(t) = t^2 - t^3 + 25t^{-5} \]
Continuous Time Signal

- Types of continuous time signals
  - Periodic or Non-periodic

\[
x(t + \Delta t) = x(t)
\]
Continuous Time Signal

- Types of continuous time signals
  - Periodic or Non-periodic
  - Real or Complex

Radar signals are complex
Continuous, Linear, Time Invariant Systems

- **Continuous**
  - If \( x(t) \) and \( y(t) \) are continuous time functions, the system is a continuous time system

- **Linear**
  - If the system satisfies \( T[\alpha x_1(t) + \beta x_2(t)] = \alpha y_1(t) + \beta y_2(t) \)

- **Time Invariant**
  - If a time shift in the input causes the same time shift in the output

\[
x(t) \xrightarrow{\text{Continuous Linear Time Invariant System}} y(t) \quad y(t) = T[x(t)] \quad T = \text{Operator}
\]
Linear Time Invariant Systems
(Delta Function)

\[ x(t) \xrightarrow{\text{Continuous Linear Time Invariant System}} y(t) \xrightarrow{\delta(t)} \xrightarrow{\text{Continuous Linear Time Invariant System}} h(t) \]

Properties of Delta Function

\[
\begin{align*}
\delta(t) &= 0 \quad t \neq 0 \\
\delta(t) &= \infty \quad t = 0 \\
\int_{-\infty}^{\infty} \delta(t) \, dt &= 1
\end{align*}
\]

- The impulse response \( h(t) \) is the response of the system when the input is \( \delta(t) \)
Continuous Linear Time Invariant System

\[ x(t) \rightarrow y(t) \quad \delta(t) \rightarrow h(t) \]

Definition: Convolution of Two Functions

\[
x_1(t) * x_2(t) \equiv \int_{-\infty}^{\infty} x_1(\tau)x_2(t - \tau) d\tau
\]

Reversed and Shifted
Linear Time Invariant Systems

\[ y(t) = x(t) \ast h(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau \]

- The output of any continuous time, linear, time-invariant (LTI) system is the convolution of the input \( x(t) \) with the impulse response of the system \( h(t) \)
Why not Analog Sensors and Calculation Systems?

Disadvantages

- Measurement Repeatability
- Environmental Sensitivity
- Size
- Complexity
- Cost

Voltmeter

Slide Rule

Torpedo Data Computer (1940s)

Courtesy of Hannes Grobe

Courtesy of oschene

Courtesy of US Navy
Outline

• Continuous Signals and Systems

• Sampled Data and Discrete Time Systems
  – General properties
  – A/D Conversion
  – Sampling Theorem and Aliasing
  – Convolution of Discrete Time Signals
  – Fourier Properties of Signals
    Continuous vs. Discrete
    Periodic vs. Aperiodic

• Discrete Fourier Transform (DFT)

• Fast Fourier Transform (FFT)

• Finite Impulse Response (FIR) Filters

• Weighting of Filters
Sampled Data Systems

- Digital signal processing deals with sampled data

- Digital processing differs from processing continuous (analog) signals

- Digital Samples are obtained with a “Sample and Hold” (S/H) Amplifier followed by an “Analog-to-Digital” (A/D) converter
  - Sampling rate
  - Word length
• Sampling converts a continuous signal into a sequence of numbers

- Continuous-time System
- Discrete-time System

• Radar signals are complex
Outline

- Continuous Signals and Systems
- Sampled Data and Discrete Time Systems
  - General properties
  - A/D Conversion
  - Sampling Theorem and Aliasing
  - Convolution of Discrete Time Signals
  - Fourier Properties of Signals
    - Continuous vs. Discrete
    - Periodic vs. Aperiodic
- Discrete Fourier Transform (DFT)
- Fast Fourier Transform (FFT)
- Finite Impulse Response (FIR) Filters
- Weighting of Filters
Ideal Analog to Digital (A/D) Converter

\[ V_{\text{OUT}} = \frac{-V_{\text{FS}}}{2} + \frac{V_{\text{FS}}}{2} \cdot \left( \frac{q}{2} - \frac{q}{2} \right) \]

\[ V_{\text{ERROR}} = V_{\text{OUT}} - V_{\text{IN}} \]

\[ \sigma_{V_{\text{ERROR}}}^2 = \frac{q}{12} \]
“Non-Perfect Nature” of A/D Converters

- Gain
- Missing bits
- Monotonicity
- Offset
- Nonlinearity
- Missing bits
Single Tone A/D Converter Testing

For Ideal A/D \( S/N = 6.02N + 1.76 \text{ dB} \)
A/D Word Length

• A / D output is signed N bit integers
  – Twos complement arithmetic
  – Quantization noise power = 1/12

• Signal-to-noise ratio \( \left( \frac{S^2}{N_0} \right) \), must fit within the word length:
  – \( S^2 \) = maximum signal power (target, jamming, clutter)
  – \( N_0 \) = thermal noise power in A / D input

\[
2^{L-1} > \alpha S \quad 1/12 < N_0
\]
  – Typically, \( \alpha \approx 4 \) to reduce clipping (limiting)

• Required word length: \( L > \left( \frac{\text{SNR}_{\text{DB}}}{6} \right) + 1.2 \)

\[
\text{SNR}_{\text{DB}} = 10\log_{10} \text{SNR}
\]
Outline

- Continuous Signals and Systems
- Sampled Data and Discrete Time Systems
  - General properties
  - A/D Conversion
  - Sampling Theorem and Aliasing
  - Convolution of Discrete Time Signals
  - Fourier Properties of Signals
    - Continuous vs. Discrete
    - Periodic vs. Aperiodic
- Discrete Fourier Transform (DFT)
- Fast Fourier Transform (FFT)
- Finite Impulse Response (FIR) Filters
- Weighting of Filters
Waveform Sampling

• Sampling converts a continuous signal into a sequence of numbers

![Diagram showing continuous-time and discrete-time systems]

• Radar signals are complex
• Sampling Theorem constraint (a.k.a. Nyquist criterion) to prevent “aliasing”:
  – For continuous aperiodic signals:

\[ F_s \geq 2B \quad F_s = \text{Sampling Frequency} \]

• Nyquist criterion:
  – Permits reconstruction via a low pass filtering
  – Eliminates Aliasing
Signal Sampling Issues

- Signal Reconstruction

\[ F_s > 2B \]

\[ X(F) \]

\[ \text{LPF} \]

\[ 0 \quad F_s \quad 2F_s \]

\[ X_c(F) \]

- Elimination of “Aliasing”

\[ F_s < 2B \]

\[ \text{Overlapping, Aliased Spectra} \]

\[ -F_s \quad 0 \quad F_s \quad 2F_s \quad 3F_s \quad 4F_s \]
The Sampling Theorem

• If $x_c(t)$ is strictly band limited,

$$X(F) = 0 \quad \text{for} \quad |F| > B$$

then, $x_c(t)$ may be uniquely recovered from its samples $x[n]$ if

$$F_S = \frac{2\pi}{T_S} \geq 2B$$

The frequency $B$ is called the **Nyquist frequency**, and the minimum sampling frequency, $F_S = 2B$, is called the **Nyquist rate**.
Spectrum of a Sampled Signal

- Sampling periodically replicates the spectrum
  - Fourier transform of a sampled signal is periodic

- If $X_c(F)$ and $X(F)$ are the spectra of $x_c(t)$ and $x[n]$

\[
X_c(F) = \int_{-\infty}^{\infty} x_c(t)e^{-j2\pi F t}dt
\]

\[
X(F) = \int \left( \sum_{n=-\infty}^{\infty} g(t)\delta(t - nT) \right)e^{-j2\pi F t}dt
\]

\[
= \sum_{n=-\infty}^{\infty} x[n]e^{-j2\pi nF/F_s}
\]
Distortion of a Signal Spectrum by “Aliasing”

- Assume $x_c(t)$ band limited so that $X(f) = 0$, for $|f| > B$

- If $x_c(t)$ is sampled with $F_S \geq 2B$

- If $x_c(t)$ is sampled with $F_S < 2B$
Effect of Sampling Rate on Frequency

Continuous Signal

$$x_c(t) = e^{-At}, \quad A > 0$$

Its Fourier Transform

$$X_c(F) = \frac{2A}{A^2 + (2\pi F)^2}$$

Sampled Signal

$$x[n] = x_c(nT) = e^{-A|n|} = (e^{-AT})^{|n|} = a^n$$

Its Fourier Transform

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n} = \frac{1 - a^2}{1 - 2a \cos \omega + a^2}, \quad \omega = 2\pi \frac{F}{F_s}$$

Reconstructed Signal

$$\hat{x}_c(t) \rightarrow \text{Inverse Fourier Transform}$$

Adapted from Proakis and Manolakis, Reference 1
Spectrum of Reconstructed Signal

Continuous Signal

\[ x_c(t) \]

Frequency Spectrum

\[ X_c(F) \]

Sampled Signal

\[ x[n] = x_c(nT) \]

\[ T = \frac{1}{3} \text{ sec} \]

Sampled Signal

\[ x[n] = x_c(nT) \]

\[ T = 1 \text{ sec} \]

Adapted from Proakis and Manolakis, Reference 1
Outline

• Continuous Signals and Systems

• Sampled Data and Discrete Time Systems
  – General properties
  – A/D Conversion
  – Sampling Theorem and Aliasing
  – Convolution of Discrete Time Signals
  – Fourier Properties of Signals
    Continuous vs. Discrete
    Periodic vs. Aperiodic

• Discrete Fourier Transform (DFT)

• Fast Fourier Transform (FFT)

• Finite Impulse Response (FIR) Filters

• Weighting of Filters
Convolution for Discrete Time Systems

Continuous-time System

\[ x(t) \rightarrow y(t) \]
\[ y(t) = \int_{-\infty}^{\infty} h(\tau)x(t - \tau)\,d\tau \]

Discrete-time System

\[ x[n] \rightarrow y[n] \]
\[ y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n - k] \]
Graphical Implementation of Convolution

\[ y[n] = \sum_{k=-\infty}^{\infty} h[k] x[n-k] = \sum_{k=-\infty}^{\infty} x[k] h[n-k] \]

Example:

\[ h[k] = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \quad x[k] = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \end{bmatrix} \]

- Step 1: Plot the sequences, \( x[k] \) and \( h[k] \)
Graphical Implementation of Convolution

\[ y[n] = \sum_{k=-\infty}^{\infty} h[k] x[n-k] = \sum_{k=-\infty}^{\infty} x[k] h[n-k] \]

Example:

\[ h[k] = [1, 2, 3] \quad x[k] = [1, 2, 3, 4, 5] \quad h[-k] = [3, 2, 1] \]

- Step 2: Take one of the sequences and time reverse it
Graphical Implementation of Convolution

\[ y[n] = \sum_{k=-\infty}^{\infty} h[k] x[n-k] = \sum_{k=-\infty}^{\infty} x[k] h[n-k] \]

Example:

\[ h[k] = \begin{array}{cccc} 1 & 2 & 3 \\ 0 & 1 & 2 \end{array} \quad x[k] = \begin{array}{cccc} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 \end{array} \quad h[-k] = \begin{array}{c} 3 \\ -2 \ -1 \ 0 \end{array} \]

- Step 3: Shift \( h[-k] \) by \( n \), yielding
  - \( n < 0 \) a shift to the left
  - \( n > 0 \) a shift to the right

\[ h[n-k] = \begin{array}{cccc} 3 & 2 & 1 \\ n-2, n-1, n \end{array} \]
Graphical Implementation of Convolution

\[ y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] \]

Example:

- Step 4: For each value of \( n \), multiply the sequences \( x[k] \) and \( h[-k] \); and add products together for all values of \( k \) to produce \( y[n] \).
Graphical Implementation of Convolution

\[ y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] \]

Example:

\[ h[k] = \begin{cases} 3 & k = 1 \\ 2 & k = 2 \\ 1 & k = 0 \end{cases} \quad x[k] = \begin{cases} 4 & k = 1 \\ 3 & k = 2 \\ 1 & k = 0 \end{cases} \quad h[-k] = \begin{cases} 3 & k = 0 \\ 2 & k = 1 \\ 1 & k = 2 \end{cases} \]

for \( n = 0 \)

No overlap – \( y[n] = 0 \)
Graphical Implementation of Convolution

\[ y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] \]

Example:

\[ h[k] = 1 \quad 2 \quad 3 \]
\[ x[k] = 1 \quad 2 \quad 3 \quad 4 \quad 5 \]

\[ h[1-k] = 3 \quad 2 \quad 1 \]

One sample overlaps – \[ y[n] = (1 \times 1) = 1 \]
Graphical Implementation of Convolution

\[ y[n] = \sum_{k=-\infty}^{\infty} h[k] x[n-k] = \sum_{k=-\infty}^{\infty} x[k] h[n-k] \]

Example:

\[ h[k] = \begin{cases} 3 & \text{for } k = 1 \\ 2 & \text{for } k = 2 \\ 1 & \text{for } k = 0 \\ 0 & \text{otherwise} \end{cases} \]

\[ x[k] = \begin{cases} 4 & \text{for } k = 4 \\ 3 & \text{for } k = 3 \\ 2 & \text{for } k = 2 \\ 1 & \text{for } k = 1 \\ 0 & \text{otherwise} \end{cases} \]

\[ h[2-k] = \begin{cases} 3 & \text{for } k = 1 \\ 2 & \text{for } k = 0 \\ 1 & \text{for } k = 1 \\ 0 & \text{otherwise} \end{cases} \]

Two samples overlaps – \( y[n] = (1 \times 2) + (2 \times 1) = 4 \)
Graphical Implementation of Convolution

\[ y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] \]

Example:

\[ h[k] = \begin{cases} 3 & \text{for } k = 0 \\ 2 & \text{for } k = 1, 2 \end{cases} \quad x[k] = \begin{cases} 4 & \text{for } k = 1, 2, 3 \\ 1 & \text{for } k = 4 \end{cases} \]

for \( n = 3 \)

\[ h[3-k] = \begin{cases} 3 & \text{for } k = 1, 2, 3 \\ 2 & \text{for } k = 0 \\ 1 & \text{for } k = 4 \end{cases} \]

Three samples overlap – \( y[n] = (1 \times 3) + (2 \times 2) + (4 \times 1) = 11 \)
Graphical Implementation of Convolution

\[ y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] \]

Example:

\[ h[k] = \begin{cases} 1 & \text{for } k=0,1,2 \\ 2 & \text{for } k=3 \end{cases} \]
\[ x[k] = \begin{cases} 1 & \text{for } k=1,2,3,4 \\ 2 & \text{for } k=5 \end{cases} \]

for \( n = 4 \)

\[ h[4-k] = \begin{cases} 1 & \text{for } k=2,3,4,5 \end{cases} \]

Three samples overlaps – \( y[n] = (2 \times 3) + (4 \times 2) + (3 \times 1) = 17 \)
Graphical Implementation of Convolution

\[ y[n] = \sum_{k=\infty}^{\infty} h[k]x[n-k] = \sum_{k=\infty}^{\infty} x[k]h[n-k] \]

Example:

\[ h[k] = \begin{cases} 
1 & \text{for } k = 0, 1, 2 \\
2 & \text{for } k = 1, 2, 3 \\
3 & \text{for } k = 2, 3, 4 \\
0 & \text{otherwise} 
\end{cases} \]

\[ x[k] = \begin{cases} 
1 & \text{for } k = 1, 2, 3 \\
2 & \text{for } k = 1, 2, 3 \\
3 & \text{for } k = 2, 3, 4 \\
0 & \text{otherwise} 
\end{cases} \]

\[ h[5-k] = \begin{cases} 
3 & \text{for } k = 3, 4, 5 \\
2 & \text{for } k = 2, 3, 4 \\
1 & \text{for } k = 1, 2, 3 \\
0 & \text{otherwise} 
\end{cases} \]

Three samples overlaps – \( y[n] = (4\times 3) + (3\times 2) + (1\times 1) = 17 \)
Graphical Implementation of Convolution

\[ y[n] = \sum_{k=\infty}^{\infty} h[k]x[n-k] = \sum_{k=\infty}^{\infty} x[k]h[n-k] \]

Example:

\[ h[k]= \]

\[ x[k]= \]

\[ h[6-k]= \]

Two samples overlaps – \( y[n] = (3x3)+(1x2) = 11 \)
Graphical Implementation of Convolution

\[ y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] \]

Example:

\[ h[k] = \begin{align*} 
1 & \quad 2 & \quad 3 \\
0 & \quad 1 & \quad 2 
\end{align*} \]

\[ x[k] = \begin{align*} 
1 & \quad 2 & \quad 3 & \quad 4 & \quad 5 \\
1 & \quad 2 & \quad 3 & \quad 4 & \quad 5 
\end{align*} \]

\[ h[7-k] = \begin{align*} 
3 & \quad 2 & \quad 1 \\
5 & \quad 6 & \quad 7 
\end{align*} \]

Output of Convolution

One sample overlaps – \( y[n] = (1\times3) = 3 \)
Graphical Implementation of Convolution

\[ y[n] = \sum_{k=-\infty}^{\infty} h[k] x[n-k] = \sum_{k=-\infty}^{\infty} x[k] h[n-k] \]

Example:

\[ h[k] = \begin{array}{cccc}
0 & 1 & 2 & 3 \\
4 & 3 & 2 & 1 \\
\end{array} \quad x[k] = \begin{array}{cccc}
1 & 2 & 3 & 4 \\
5 & 4 & 3 & 2 \\
\end{array} \quad h[8-k] = \begin{array}{cccc}
3 & 2 & 1 & 0 \\
6 & 7 & 8 & 0 \\
\end{array} \]

for \( n = 8 \)

No overlap - \( y[n] = 0 \)
Summary - Linear Discrete Time Systems

• Any Linear and Time-Invariant (LTI) system can be completely described by its impulse response sequence

\[ \delta[n] \rightarrow h[n] \]

• The output of any LTI can be determined using the convolution summation

\[ y[n] = \sum_{k=\infty}^{\infty} h[k] x[n-k], \quad -\infty < n < \infty \]

• The impulse response provides the basis for the analysis of an LTI system in the time-domain

• The frequency response function provides the basis for the analysis of an LTI system in the frequency-domain

Adapted from MIT LL Lecture Series by D. Manolakis
Outline

• Continuous Signals and Systems

• Sampled Data and Discrete Time Systems
  – General properties
  – A/D Conversion
  – Sampling Theorem and Aliasing
  – Convolution of Discrete Time Signals
  – Fourier Properties of Signals
    Continuous vs. Discrete
    Periodic vs. Aperiodic

• Discrete Fourier Transform (DFT)

• Fast Fourier Transform (FFT)

• Finite Impulse Response (FIR) Filters

• Weighting of Filters
Frequency Analysis of Signals

- Decomposition of signals into their frequency components
  - A series of sinusoids of complex exponentials

- The general nature of signals
  - Continuous or discrete
  - Aperiodic or periodic

- Radar echoes, from each transmitted pulse, are **continuous and aperiodic**, and are usually transformed into discrete signals by an A/D converter before further processing
  - Complex signals
Time and Frequency Domains

Analysis

Fourier Transform

Time History

Time Domain

Frequency Spectrum

Frequency Domain

Inverse Fourier Transform

Synthesis
Fourier Properties of Signals

- **Continuous-Time Signals**
  - Periodic Signals: Fourier Series
  - Aperiodic Signals: Fourier Transform

- **Discrete-Time Signals**
  - Periodic Signals: Fourier Series
  - Aperiodic Signals: Fourier Transform
Fourier Transform for Continuous-Time Aperiodic Signals

Time Domain
Continuous and Aperiodic Signals

\[ x(t) \]

Frequency Domain
Continuous and Aperiodic Signals

\[ X(F) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi Ft} dt \]

\[ x(t) = \int_{-\infty}^{\infty} X(F) e^{j2\pi Ft} dF \]

Adapted from Manolakis et al, Reference 1
Fourier Properties of Signals

• Continuous-Time Signals
  – Periodic Signals: Fourier Series
  – Aperiodic Signals: Fourier Transform

• Discrete-Time Signals
  – Periodic Signals: Fourier Series
  – Aperiodic Signals: Fourier Transform
Fourier Transform for Discrete-Time Aperiodic Signals

Time Domain
Discrete and Aperiodic Signals

Frequency Domain
Continuous and Periodic Signals

\[ X(\omega) = \sum_{n=-\infty}^{\infty} X[n] e^{-j\omega n} \]

\[ x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega) e^{j\omega n} d\omega \]

Adapted from Malolakis et al, Reference 1
# Summary of Time to Frequency Domain Properties

<table>
<thead>
<tr>
<th>Continuous- Time Signals</th>
<th>Discrete- Time Signals</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Time-Domain</strong></td>
<td><strong>Frequency-Domain</strong></td>
</tr>
<tr>
<td>( x(t) )</td>
<td>( x[n] )</td>
</tr>
<tr>
<td>( c_k = \frac{1}{T_p} \int_{-T_p}^{T_p} x(t) e^{-j2\pi k F_0 t} dt )</td>
<td>( F_0 = \frac{1}{T_p} )</td>
</tr>
<tr>
<td>( x(t) = \sum_{k=\infty} ) ( c_k e^{j2\pi k F_0 t} )</td>
<td>( x[n] )</td>
</tr>
</tbody>
</table>

**Periodic Signals**
- Discrete and Periodic
- Continuous and Periodic

**Aperiodic Signals**
- Continuous and Aperiodic
- Discrete and Aperiodic

**Fourier Transforms**
- Continuous and Aperiodic
- Discrete and Aperiodic

Adapted from Proakis and Manolakis, Reference 1
Outline

• Continuous Signals and Systems

• Sampled Data and Discrete Time Systems

• Discrete Fourier Transform (DFT)
  – Calculation

• Fast Fourier Transform (FFT)

• Finite Impulse Response (FIR) Filters

• Weighting of Filters
Direct DFT Computation

\[ X[k] = \sum_{n=0}^{N-1} x[n] W_{kn} \quad 0 \leq k \leq N - 1 \]

\[ W_{kn} = e^{-2\pi j k n / N} \]

\[ X_R[k] = \sum_{n=0}^{N-1} \left\{ x_R[n] \cos\left(\frac{2\pi}{N} kn\right) + x_I[n] \sin\left(\frac{2\pi}{N} kn\right) \right\} \]

\[ X_I[k] = -\sum_{n=0}^{N-1} \left\{ x_R[n] \sin\left(\frac{2\pi}{N} kn\right) - x_I[n] \cos\left(\frac{2\pi}{N} kn\right) \right\} \]

- 1. \(2N^2\) evaluations of trigonometric functions \(\approx N^2\) Complex MADS
- 2. \(4N^2\) real (\(N^2\) complex) multiplications MADS Multiply And Divides
- 3. \(4N(N-2)\) real (\(N(N-1)\) complex) additions
- 4. A number of indexing and addressing operations

Adapted from MIT LL Lecture Series by D. Manolakis
Outline

- Continuous Signals and Systems
- Sampled Data and Discrete Time Systems
- Discrete Fourier Transform (DFT)
- Fast Fourier Transform (FFT)
- Finite Impulse Response (FIR) Filters
- Weighting of Filters
Fast Fourier Transform (FFT)

- An algorithm for each efficiently computing the Discrete Fourier Transform (DFT) and its inverse

- DFT $O\left(N^2\right)$ MADS (Multiplies and Divides)

- FFT $O\left(\frac{N\log_2 N}{2}\right)$ MADS

- FFT algorithm Development - Cooley / Tukey (1965) Gauss (1805)

- Many variations and efficiencies of the FFT algorithm exist
  - Decimation in Time (input - bit reversed, output - natural order)
  - Decimation in Frequency (input - natural order, output - bit reversed)

- The FFT calculation is broken down into a number of sequential stages, each stage consisting of a number of relatively small calculations called “Butterflies”
Radix 2 Decimation in Time FFT Algorithm

\[ X[k] = \sum_{n=0}^{N-1} x[n] e^{-2\pi j k n/N} = \sum_{n=0}^{N-1} x[n] W_N^{kn} \quad 0 \leq k \leq N - 1 \quad W_N^{kn} = e^{-2\pi j k n/N} \]

- Divide DFT of size N into two interleaved DFTs, each of size N/2
  - Example will be \( N = 2^3 = 8 \)
  - Input to each DFT are even and odd \( x[n] \)s, respectively
- Solve each stage recursively, until the size of the stage’s DFT is 2.

\[ X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk} = \sum_{n \text{ Even}} x[n] W_N^{nk} + \sum_{n \text{ Odd}} x[n] W_N^{nk} \]

\[ = \sum_{l=0}^{N/2-1} g[l] W_N^{lk} + \sum_{l=0}^{N/2-1} h[l] W_N^{(2l+1)k} = \sum_{l=0}^{N/2-1} g[l] W_N^{1k} + W_N^{k} \sum_{l=0}^{N/2-1} h[l] W_N^{lk} \]

Even index and odd index terms of \( x[n] \)

\[ \text{N/2 point DFT of } g[l] = G[k] \]

\[ \text{N/2 point DFT of } h[l] = H[k] \]
Radix 2 Decimation in Time FFT Algorithm (continued)

\[ X[k] = G[k] + W_{N}^{nk} H[k] \]

- Using the periodicity of the complex exponentials:
  
  \[ G[k] = G\left[ k + \frac{N}{2} \right] \quad H[k] = H\left[ k + \frac{N}{2} \right] \]

- And the following properties of the “twiddle factors”:
  
  \[ W_{N}^{k+(N/2)} = W_{N}^{k} W_{N}^{N/2} = -W_{N}^{k} \]
  
  then \[ W_{N}^{k+(N/2)} H\left( k + \left( \frac{N}{2} \right) \right) = -W_{N}^{k} H(k) \]

- A block diagram of this computational flow is graphically illustrated in the next chart for an 8 point FFT
8 Point Decimation in Time FFT Algorithm (After First Decimation)

\[ x[0] \rightarrow G[0] \rightarrow W_8^0 x[0] \]
\[ x[1] \rightarrow H[0] \rightarrow W_8^4 x[4] \]
Decimation of 4 Point into two 2 point DFTs

• If N/2 is even, \( g[n] \) and \( h[n] \) may again be decimated

\[
G[k] = \sum_{n=0}^{N/2-1} g[n] W_{N/2}^{nk} = \sum_{n \text{ Even}} g[n] W_{N/2}^{nk} + \sum_{n \text{ Odd}} g[n] W_{N/2}^{nk}
\]

• This leads to:

\[
G[k] = \sum_{n=0}^{N/4-1} g[2n] W_{N/4}^{nk} + W_k^{N/2} \sum_{n=0}^{N/2-1} g[2n+1] W_{N/4}^{nk}
\]
Butterfly for 2 Point DFT

\[
\begin{align*}
q[0] + q[1] &= Q[0] = q[0] + q[1] \\
\end{align*}
\]

Now, Putting it all together.....
Flow of 8-Point FFT
(Radix 2 - Decimation in Time Algorithm)
Basic FFT Computation Flow Graph

- Each "Butterfly" takes 2 MADS (Multiplies and Adds)
- Twiddle Factors (For 8 point FFT)
  - $W_8^0 = e^{-0} = 1$
  - $W_8^1 = e^{-2\pi j/8} = e^{-\pi j/4} = (1 - j)/\sqrt{2}$
  - $W_8^2 = e^{-\pi j/2} = -j$
  - $W_8^3 = e^{-3\pi j/4} = (-1 - j)/\sqrt{2}$

- 12 Butterflies implies 12 MADS vs. 64 MADS for 8 point DFT
- 512 point FFT more than 100 times faster than 512 DFT
Computational Speed – DFT vs. FFT

- Discrete Fourier Transform \( (O \sim N^2) \)
- Fast Fourier Transform \( (O \sim N \log_2 N) \)

Adapted from Lyons, Reference 2

Number of Complex Multiplications

Number of points in Radix 2 FFT

Lines Drawn Through Data Points

Drawn Through Data Points
• Fast Fourier Transform (FFT) algorithms make possible the computation of DFT with \( O \left( \frac{N}{2} \log_2 N \right) \text{MADS} \) as opposed to \( O N^2 \text{MADS} \).

• Many other implementations of the FFT exist:
  – Radix 2 decimation in frequency algorithm
  – Radar-Brenner algorithm
  – Bluestein’s algorithm
  – Prime Factor algorithm

• The details of FFT algorithms are important to the designers of real-time DSP systems in software or hardware.

• An interesting history of FFT algorithms
Outline

- Continuous Signals and Systems
- Sampled Data and Discrete Time Systems
- Discrete Fourier Transform (DFT)
- Fast Fourier Transform (FFT)
- Finite Impulse Response (FIR) Filters
- Weighting of Filters
Finite and Infinite Response Filters

- **Infinite Impulse Response (IIR) Filters**
  - Output of filter depends on past time history \((-\infty)\)
  - Example:
    \[ y[n] = \frac{1}{M} x[n] + \frac{M-1}{M} y[n-1] \]

- **Finite Impulse Response (FIR) Filters**
  - Output depends on the finite past
  - Example: DFT
    \[ X[k] = \sum_{n=0}^{N-1} x[n] e^{-2\pi j k n/N} \]
  - Other examples:
    \[ y[k] = \sum_{n=0}^{N-1} a[k,n] x[n] x[1] \]
    or \[ y[n] = x[n,2] - x[n,1] \]
Four Basic Filter Types - An Idealization

**Ideal Low Pass Filter**

\[ |H(e^{j\omega})| \]

\[ -\pi \quad -\omega_c \quad \omega_c \quad \pi \]

**Ideal High Pass Filter**

\[ |H(e^{j\omega})| \]

\[ -\pi \quad -\omega_c \quad \omega_c \quad \pi \]

**Ideal Bandpass Filter**

\[ |H(e^{j\omega})| \]

\[ -\pi \quad -\omega_2 - \omega_1 \quad \omega_1 \quad \omega_2 \quad \pi \]

**Ideal Bandstop Filter**

\[ |H(e^{j\omega})| \]

\[ -\pi \quad -\omega_2 - \omega_1 \quad \omega_1 \quad \omega_2 \quad \pi \]
Outline

- Continuous Signals and Systems
- Sampled Data and Discrete Time Systems
- Discrete Fourier Transform (DFT)
- Fast Fourier Transform (FFT)
- Finite Impulse Response (FIR) Filters
- Weighting of Filters
Windowing / Weighting of Filters

• If we take a square pulse, sample it M times, and calculate the Fourier transform of this uniform rectangular “window”:

\[ W(\omega) = \sum_{n=0}^{M-1} e^{-j \omega n} = \frac{1 - e^{-j \omega M}}{1 - e^{-j \omega}} = e^{-j(M-1)/2} \frac{\sin(\omega M / 2)}{\sin(\omega / 2)} \]

\[ |W(\omega)| = \frac{|\sin(\omega M / 2)|}{|\sin(\omega / 2)|} \quad -\pi \leq \omega \leq \pi \]

• This is recognized as the sinc function which has 13 dB sidelobes

• If lower sidelobes are needed, at the cost of a widened pass band, one can multiply the elements of the pulse sequence with one of a number of weighting functions, which will adjust the sidelobes appropriately
Commonly Used Window Functions

- **Rectangular**
  \[ w[n] = \begin{cases} 
  1, & 0 \leq n \leq M \\
  0, & \text{otherwise} 
  \end{cases} \]

- **Bartlett (triangular)**
  \[ w[n] = \begin{cases} 
  2n / M, & 0 \leq n \leq M / 2 \\
  2 - 2n / M, & M / 2 < n \leq M \\
  0, & \text{otherwise} 
  \end{cases} \]

- **Hanning**
  \[ w[n] = \begin{cases} 
  0.5 - 0.5 \cos \left( 2 \pi \frac{n}{M} \right), & 0 \leq n \leq M \\
  0, & \text{otherwise} 
  \end{cases} \]

- **Hamming**
  \[ w[n] = \begin{cases} 
  0.54 - 0.46 \cos \left( 2 \pi \frac{n}{M} \right), & 0 \leq n \leq M \\
  0, & \text{otherwise} 
  \end{cases} \]

- **Blackman**
  \[ w[n] = \begin{cases} 
  0.42 - 0.5 \cos \left( 2 \pi \frac{n}{M} \right) + 0.08 \cos \left( 4 \pi \frac{n}{M} \right), & 0 \leq n \leq M \\
  0, & \text{otherwise} 
  \end{cases} \]
## Comparison of Common Windows

<table>
<thead>
<tr>
<th>Type of Window</th>
<th>Peak Sidelobe Amplitude (dB)</th>
<th>Approximate Width of Main Lobe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rectangular</td>
<td>−13</td>
<td>$4\pi/(M + 1)$</td>
</tr>
<tr>
<td>Bartlett (triangular)</td>
<td>−25</td>
<td>$8\pi / M$</td>
</tr>
<tr>
<td>Hanning</td>
<td>−31</td>
<td>$8\pi / M$</td>
</tr>
<tr>
<td>Hamming</td>
<td>−41</td>
<td>$8\pi / M$</td>
</tr>
<tr>
<td>Blackman</td>
<td>−57</td>
<td>$12\pi / M$</td>
</tr>
</tbody>
</table>
Comparison of Rectangular & Hamming Windows

\[ 20 \log_{10}|W(\omega)| \]

Normalized frequency \((f = F/F_s)\)
Summary

• A brief review of the prerequisite Signal & Systems, and Digital Signal Processing knowledge base for this radar course has been presented
  – Viewers requiring a more in depth exposition of this material should consult the references at the end of the lecture

• The topics discussed were:
  – Continuous signals and systems
  – Sampled data and discrete time systems
  – Discrete Fourier Transform (DFT)
  – Fast Fourier Transform (FFT)
  – Finite Impulse Response (FIR) filters
  – Weighting of filters
References


Acknowledgements

- Dr Dimitris Manolakis
- Dr. Stephen C. Pohlig
- Dr William S. Song
Homework Problems

• From Proakis and Manolakis, Reference 1
  – Problems 2.1, 2.17, 4.9a and b, 4.10 a and b, 6.1, 6.9 a and b, 8.1 and 8.8

• Or

• And from Hays, Reference 4
  – Problems 1.41, 1.49, 1.54, 1.59, 2.46, 2.57, 2.58, 3.27, 3.28, 3.34, 6.44, 6.45