Fall 2012, EE123 Digital Signal Processing
Lecture 5

Miki Lustig, UCB

September 4, 2012

Motivations for Discrete Fourier Transform

Sampled representation in time and frequency

- Numerical Fourier analysis requires a Fourier representation that is sampled in time and frequency
- Sampling in one domain corresponds to periodicity in the other domain
- Hence, we want a representation that is discrete and period in both *time* and *frequency*.
- This is the discrete-time Fourier series or, equivalently, the discrete Fourier transform.
- However, we are often dealing with signals that are *not* periodic, but still using the DFT. This requires special considerations.

Motivations for Discrete Fourier Transform

Efficient Implementions

Very efficient implementations of the discrete Fourier transform exist

- Direct evaluation of DFT: $O(N^2)$
- Fast Fourier Transform (FFT) algorithms: $O(N \log_2 N)$
- FFT algorithms are most straightforward for $N = 2^m$
- MATLAB commands:
  - `X=fft(x)`
  - `x=ifft(X)`

Convolution can be implemented efficiently using the FFT

- Direct convolution: $O(N^2)$
- FFT-based convolution: $O(N \log_2 N)$

Discrete Fourier Series

Definition

- Consider N-periodic signal:

$$\tilde{x}[n+N] = \tilde{x}[n] \quad \forall n$$

and its frequency-domain representation, which is also N-periodic:

$$\tilde{X}[k+N] = \tilde{X}[k] \quad \forall k$$

The "~" will indicate a periodic signal or spectrum.

Discrete Fourier Series

- Define
$$W_N \triangleq e^{-j2\pi/N}$$
- $\tilde{x}[n]$ and $\tilde{X}[k]$ are related by the *discrete Fourier series*:

$$\tilde{x}[n] = \frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}[k]W_N^{-kn}$$
$$\tilde{X}[k] = \sum_{n=0}^{N-1}\tilde{x}[n]W_N^{kn}$$

Discrete Fourier Transform

- By convention, work with one period of $\tilde{x}[n]$ and $\tilde{X}[k]$:

$$x[n] \triangleq \begin{cases}\tilde{x}[n] & 0 \le n \le N-1 \\ 0 & \text{otherwise}\end{cases}$$
$$X[k] \triangleq \begin{cases}\tilde{X}[k] & 0 \le k \le N-1 \\ 0 & \text{otherwise}\end{cases}$$

From these, if desired, we can recover the periodic representations:

$$\tilde{x}[n] = x[((n))_N] = \sum_{r=-\infty}^{\infty} x[n-rN]$$
$$\tilde{X}[k] = X[((k))_N] = \sum_{r=-\infty}^{\infty} X[k-rN]$$

where $((n))_N \triangleq n \bmod N$

Discrete Fourier Transform

- The *discrete Fourier transform* relates $x[n]$ and $X[k]$:

DFT

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1}X[k]W_n^{-kn} \quad \text{Inverse DFT, synthesis}$$
$$X[k] = \sum_{n=0}^{N-1}x[n]W_n^{kn} \quad \text{DFT, analysis}$$

- Although not stated it explicitly, it is understood that

$$x[n] = 0 \quad \text{outside } 0 \le n \le N-1$$
$$X[k] = 0 \quad \text{outside } 0 \le k \le N-1$$

Alternative transform not in the book:

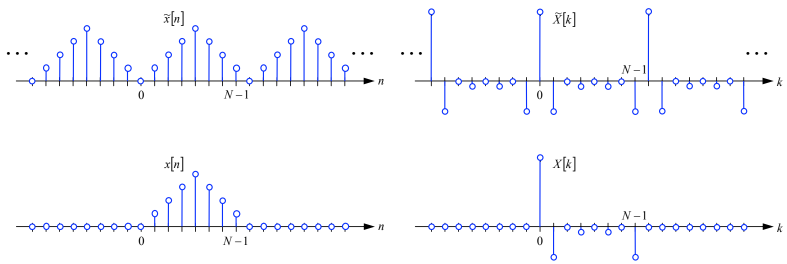
Orthonormal DFT

$$x[n] = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}X[k]W_n^{-kn} \quad \text{Inverse DFT, synthesis}$$
$$X[k] = \frac{1}{\sqrt{N}}\sum_{n=0}^{N-1}x[n]W_n^{kn} \quad \text{DFT, analysis}$$

Why use this or the other?

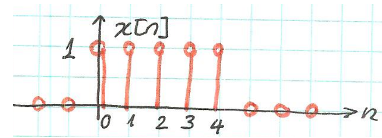
DFS vs DFT

- This figure compares the periodic signal and its DFS, $\tilde{x}[n] \overset{\text{DFS}}{\leftrightarrow} \tilde{X}[k]$, to the corresponding one-period signal and its DFT, $x[n] \overset{\text{DFT}}{\leftrightarrow} X[k]$



DFT Continued

- Example:



Take $N = 5$

$$X[k] = \begin{cases} \sum_{n=0}^{4} W_4^{nk} & k = 0,1,2,3,4 \\ 0 & \text{otherwise} \end{cases}$$
$$= 5\delta[k]$$

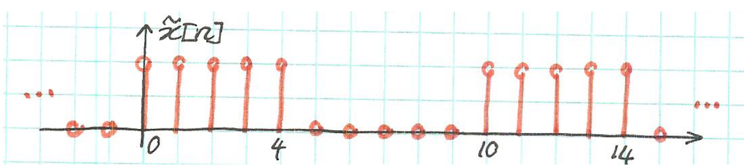
"5-point DFT "

What if we take $N = 10$?

DFT Continued

Q: What if we take $N = 10$?

A: $X[k] = \tilde{X}[k]$ where $\tilde{x}[n]$ is a period-10 sequence



Can show:

$$X[k] = \begin{cases} \sum_{n=0}^{9} W_9^{nk} & k = 0,1,2,3,4 \\ 0 & \text{otherwise} \end{cases}$$
$$= e^{-j\frac{4\pi}{10}k} \frac{\sin(\frac{\pi}{2}k)}{\sin(\frac{\pi}{10}k)}$$

"10-point DFT "

DFT vs DTFT

- *The DFT and the DTFT*

The N-point DFT of $x[n]$ is

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} = \sum_{n=0}^{N-1} x[n] e^{-j(2\pi/N)nk} \qquad 0 \le k \le N-1$$

The DTFT of $x[n]$ is

$$X(e^{j\omega}) = \sum_{n=0}^{N-1} x[n] e^{-j\omega n} \qquad -\infty < \omega < \infty$$

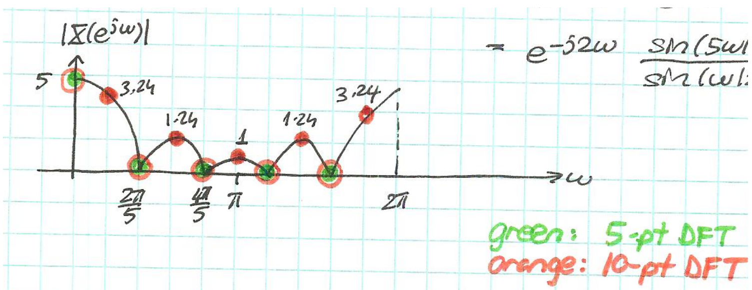
Comparing these two, we see that the DFT $X[k]$ corresponds to the DTFT $X(e^{j\omega})$ sampled at N equally spaced frequencies between 0 and 2π:

$$X[k] = X(e^{j\omega})|_{\omega = k\frac{2\pi}{N}} \qquad 0 \le k \le N-1$$

DFT vs DTFT

- Back to example:

$$X(e^{j\omega}) = \sum_{n=0}^{4} e^{-j\omega n} = e^{-j2\omega}\frac{\sin(\frac{5}{2}\omega)}{\sin(\frac{\omega}{2})}$$



DFT and IDFT

- Note that the DFT and the inverse DFT are computed in very similar fashion.

  If we write $x[n]$ as the inverse DFT of $X[k]$, multiply by N and take the complex conjugate:

$$N \cdot x^*[n] = N\left(\frac{1}{N}\sum_{k=0}^{N-1} X[k]W_N^{-kn}\right)^* = \sum_{k=0}^{N-1} X^*[k]W_N^{kn} = \mathcal{DFT}\left\{X^*[k]\right\}.$$

  However, we also know that

$$N \cdot x^*[n] = N\left(\mathcal{DFT}^{-1}\left\{X[k]\right\}\right)^*.$$

DFT and IDFT

Combining these two expressions

$$\mathcal{DFT}\left\{X^*[k]\right\} = N\left(\mathcal{DFT}^{-1}\left\{X[k]\right\}\right)^*$$

or

$$\mathcal{DFT}^{-1}\left\{X[k]\right\} = \frac{1}{N}\left(\mathcal{DFT}\left\{X^*[k]\right\}\right)^*$$

We can evaluate the inverse DFT by

- Taking the complex conjugate,
- Taking the DFT,
- Multiplying by $\frac{1}{N}$, and
- Taking the complex conjugate.

DFT as Matrix operator

- Note that the definition of the DFT and its inverse are equivalent to matrix equations

$$\begin{pmatrix} X[0] \\ \vdots \\ X[k] \\ \vdots \\ X[N-1] \end{pmatrix} = \begin{pmatrix} W_N^{00} & \cdots & W_N^{0n} & \cdots & W_N^{0(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ W_N^{k0} & \cdots & W_N^{kn} & \cdots & W_N^{k(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ W_N^{(N-1)0} & \cdots & W_N^{(N-1)n} & \cdots & W_N^{(N-1)(N-1)} \end{pmatrix} \begin{pmatrix} x[0] \\ \vdots \\ x[n] \\ \vdots \\ x[N-1] \end{pmatrix}$$

$$\begin{pmatrix} x[0] \\ \vdots \\ x[n] \\ \vdots \\ x[N-1] \end{pmatrix} = \frac{1}{N} \begin{pmatrix} W_N^{-00} & \cdots & W_N^{-0k} & \cdots & W_N^{-0(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ W_N^{-n0} & \cdots & W_N^{-nk} & \cdots & W_N^{-n(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ W_N^{-(N-1)0} & \cdots & W_N^{-(N-1)k} & \cdots & W_N^{-(N-1)(N-1)} \end{pmatrix} \begin{pmatrix} X[0] \\ \vdots \\ X[k] \\ \vdots \\ X[N-1] \end{pmatrix}$$

  This shows that straightforward computation of the N-point DFT or inverse DFT requires N^2 complex multiplies.

DFT as Matrix operator

We can write this much more compactly as a matrix equation,

$$\begin{aligned} \mathbf{X} &= \mathbf{W}_N \, \mathbf{x} \\ \mathbf{x} &= \frac{1}{N} \mathbf{W}_N^* \, \mathbf{X} \end{aligned}$$

$\mathbf{W}_N$ is the DFT coefficient matrix, and $\mathbf{x}$ and $\mathbf{X}$ are column vectors containing $x[n]$ and $X[k]$, and "$*$" is the conjugate transpose.

Note that since the columns and rows of $\mathbf{W}_N$ are orthogonal, and

$$\mathbf{W}_N \mathbf{W}_N^* = \mathbf{W}_N^* \mathbf{W}_N = N\mathcal{I}$$

where $\mathcal{I}$ is the identity matrix. Then

$$\mathbf{x} = \frac{1}{N}\mathbf{W}_N^* \mathbf{X} = \frac{1}{N}\mathbf{W}_N^* \mathbf{W}_N \mathbf{x} = \frac{1}{N}(N\mathcal{I})\,\mathbf{x} = \mathbf{x}$$

as we would expect.

Properties of Discrete Fourier Transform

These are inherited from the discrete-time Fourier series (EE120) and need not be proved

(1) Linearity

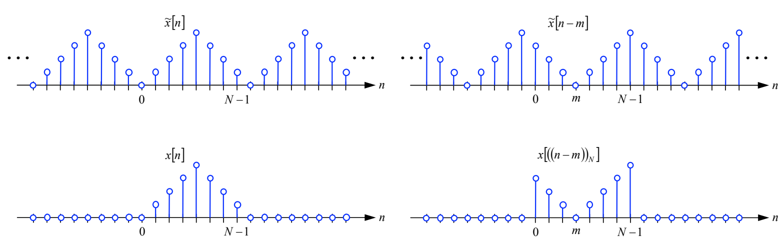
$$\alpha_1 x_1[n] + \alpha_2 x_2[n] \leftrightarrow \alpha_1 X_1[k] + \alpha_2 X_2[k]$$

(2) Circular Time Shift

$$x[((n-m))_N] \leftrightarrow X[k]e^{-j(2\pi/N)km} = X[k]W_N^{km}$$

Properties of Discrete Fourier Transform Cont.

- This figure compares a shift of the periodic sequence, $\tilde{x}[n-m]$, to a circular shift of the one-period sequence, $x[((n-m))_N]$.



Properties of Discrete Fourier Transform Cont.

(3) Circular Frequency Shift

$$x[n]e^{j(2\pi/N)nl} = x[n]W_N^{-nl} \leftrightarrow X[((k-l))_N]$$

(4) Complex Conjugation

$$x^*[n] \leftrightarrow X^*[((-k))_N]$$

(5) Time Reversal and Complex Conjugation

$$x^*[((-n))_N] \leftrightarrow X^*[k]$$

(6) Conjugate Symmetry for Real Signals

$$x[n] = x^*[n] \leftrightarrow X[k] = X^*[((-k))_N]$$

Properties of Discrete Fourier Transform Cont.

(7) Parseval's Identity

$$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N}\sum_{k=0}^{N-1} |X[k]|^2$$

Proof.

This is particularly easy using the matrix notation

$$\mathbf{x}^*\mathbf{x} = \left(\frac{1}{N}\mathbf{W}_N^*\mathbf{X}\right)^* \left(\frac{1}{N}\mathbf{W}_N^*\mathbf{X}\right) = \frac{1}{N^2}\mathbf{X}^* \underbrace{\mathbf{W}_N\mathbf{W}_N^*}_{N\cdot\mathbf{I}}\mathbf{X} = \frac{1}{N}\mathbf{X}^*\mathbf{X}$$

□

Circular Convolution Sum

- Let $x_1[n]$ and $x_2[n]$ be of length N.
- The *circular convolution* between $x_1[n]$ and $x_2[n]$ is defined as:

$$x_1[n] \circledN x_2[n] \triangleq \sum_{m=0}^{N-1} x_1[m]x_2[((n-m))_N]$$

Note that circular convolution is commutative:

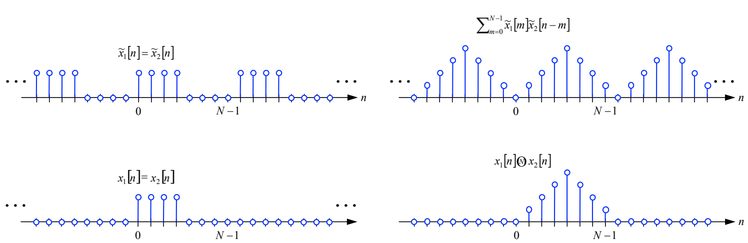
$$x_2[n] \circledN x_1[n] = x_1[n] \circledN x_2[n]$$

Circular Convolution Sum

- The circular convolution between $x_1[n]$ and $x_2[n]$ is same as one period of the periodic convolution between the corresponding periodic sequences $\tilde{x}_1[n]$ and $\tilde{x}_2[n]$:

$$x_1[n] \circledN x_2[n] = \begin{cases} \sum_{m=0}^{N-1} \tilde{x}_1[m]\tilde{x}_2[n-m] & 0 \le n \le N-1 \\ 0 & \text{otherwise} \end{cases}$$

This is illustrated in below:



Properties of Discrete Fourier Transform Cont.

(8) Circular Convolution Let $x_1[n]$ and $x_2[n]$ be of length N

$$x_1[n] \circledN x_2[n] \leftrightarrow X_1[k] \cdot X_2[k]$$

This property is very useful for DFT-based convolution.

(9) Multiplication Let $x_1[n]$ and $x_2[n]$ be of length N

$$x_1[n] \cdot x_2[n] \leftrightarrow \frac{1}{N} X_1[k] \circledN X_2[k]$$

Linear Convolution using the DFT

We start with two nonperiodic sequences:

$$x[n] \quad 0 \le n \le L-1$$
$$h[n] \quad 0 \le n \le P-1$$

We can think of $x[n]$ as a signal, and $h[n]$ as a filter inpulse response.
We want to compute the linear convolution:

$$y[n] = x[n] * h[n] = \sum_{m=0}^{L-1} x[m] * h[n-m] = \sum_{m=0}^{P-1} x[n-m]h[m]$$

$y[n] = x[n] * h[n]$ is nonzero only for $0 \le n \le L+P-2$, and is of length $L+P-1 = M$.

Linear Convolution using the DFT

We will look at two approaches for computing $y[n]$:

(1) Direct Convolution

- Evaluate the convolution sum directly.
- This requires $L \cdot P$ multiplications

(2) Using Circular Convolution

Linear Convolution using the DFT

(2) Using Circular Convolution

- Zero-pad $x[n]$ by $P-1$ zeros:

$$x_{zp}[n] = \begin{cases} x[n] & 0 \le n \le L-1 \\ 0 & L \le n \le L+P-2 \end{cases}$$

- Zero-pad $h[n]$ by $L-1$ zeros:

$$h_{zp}[n] = \begin{cases} h[n] & 0 \le n \le P-1 \\ 0 & P \le n \le L+P-2 \end{cases}$$

- Both zero-padded sequences $x_{zp}[n]$ and $h_{zp}[n]$ are of length $M = L+P-1$

Linear Convolution using the DFT

- Both zero-padded sequences $x_{zp}[n]$ and $h_{zp}[n]$ are of length $M = L+P-1$
- We can compute the linear convolution $x[n] * h[n] = y[n]$ by computing circular convolution $x_{zp}[n] \circledM h_{zp}[n]$:

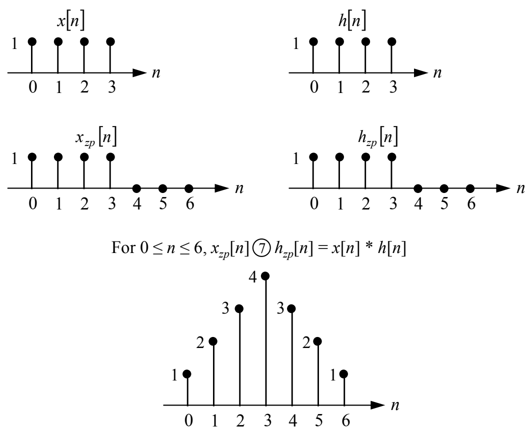
Linear convolution via circular

$$y[n] = x[n] * y[n] = \begin{cases} x_{zp}[n] \circledM h_{zp}[n] & 0 \le n \le M-1 \\ 0 & \text{otherwise} \end{cases}$$

Linear Convolution using the DFT

Example

$$L = P = 4 \qquad M = L + P - 1 = 7$$



Linear Convolution using the DFT

- In practice, the circular convolution is implemented using the DFT circular convolution property:

$$\begin{aligned} x[n] * h[n] &= x_{zp}[n] \circledM h_{zp}[n] \\ &= \mathcal{DFT}^{-1}\{DFT x_{zp}[n] \cdot \mathcal{DFT}\{h_{zp}[n]\}\} \end{aligned}$$

  for $0 \le n \le M-1$, $M = L + P - 1$.
- Advantage: This can be more efficient than direct linear convolution because the FFT and inverse FFT are $O(M \cdot \log_2 M)$.
- Drawback: We must wait until we have all of the input data. This introduces a large delay which is incompatible with real-time applications like communications.

Miki Lustig UCB. Based on Course Notes by J.M Kahn — Fall 2012, EE123 Digital Signal Processing

Linear Convolution using the DFT

- In practice, the circular convolution is implemented using the DFT circular convolution property:

$$\begin{aligned} x[n] * h[n] &= x_{zp}[n] \circledM h_{zp}[n] \\ &= \mathcal{DFT}^{-1}\{DFT x_{zp}[n] \cdot \mathcal{DFT}\{h_{zp}[n]\}\} \end{aligned}$$

  for $0 \le n \le M-1$, $M = L + P - 1$.
- Advantage: This can be more efficient than direct linear convolution because the FFT and inverse FFT are $O(M \cdot \log_2 M)$.
- Drawback: We must wait until we have all of the input data. This introduces a large delay which is incompatible with real-time applications like communications.
- Approach: Break input into smaller blocks. Combine the results using 1. *overlap and save* or 2. *overlap and add* .

Miki Lustig UCB. Based on Course Notes by J.M Kahn — Fall 2012, EE123 Digital Signal Processing

Block Convolution

Problem

An input signal $x[n]$ has very long length, which can be considered infinite.
An impulse response $h[n]$ has length P.
We want to compute the linear convolution

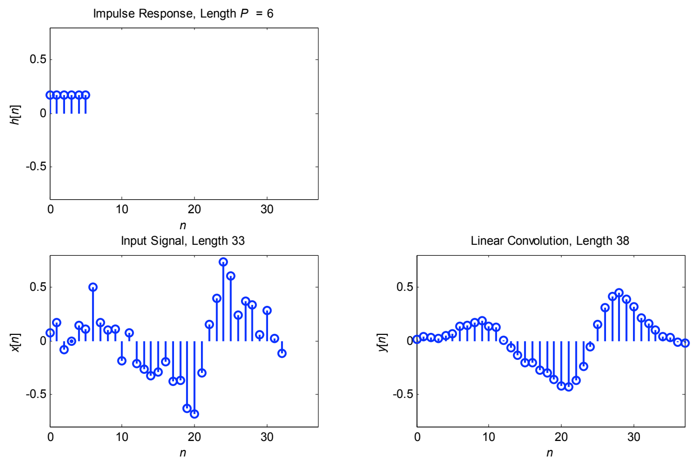
$$y[n] = x[n] * h[n]$$

using block lengths shorter than the input signal length.

Miki Lustig UCB. Based on Course Notes by J.M Kahn — Fall 2012, EE123 Digital Signal Processing

Block Convolution

Example:



Overlap-Add Method

We decompose the input signal $x[n]$ into non-overlapping segments $x_r[n]$ of length L:

$$x_r[n] = \begin{cases} x[n] & rL \leq n \leq (r+1)L - 1 \\ 0 & \text{otherwise} \end{cases}$$

The input signal is the sum of these input segments:

$$x[n] = \sum_{r=0}^{\infty} x_r[n]$$

The output signal is the sum of the output segments $x_r[n] * h[n]$:

$$y[n] = x[n] * h[n] = \sum_{r=0}^{\infty} x_r[n] * h[n] \qquad (1)$$

Each of the output segments $x_r[n] * h[n]$ is of length $N = L + P - 1$.

Overlap-Add Method

We can compute each output segment $x_r[n] * h[n]$ with linear convolution.

DFT-based circular convolution is usually more efficient:

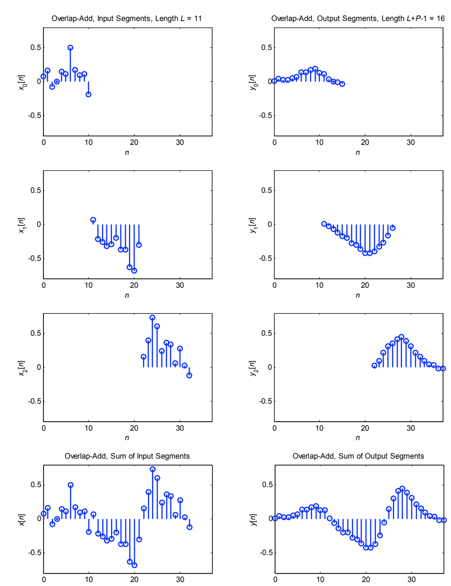
- Zero-pad input segment $x_r[n]$ to obtain $x_{r,\text{zp}}[n]$, of length N.
- Zero-pad the impulse response $h[n]$ to obtain $h_{\text{zp}}[n]$, of length N (this needs to be done only once).
- Compute each output segment using:

$$x_r[n] * h[n] = \mathcal{DFT}^{-1}\left\{\mathcal{DFT}\left\{x_{r,\text{zp}}[n]\right\} \cdot \mathcal{DFT}\left\{h_{\text{zp}}[n]\right\}\right\}$$

Since output segment $x_r[n] * h[n]$ starts offset from its neighbor $x_{r-1}[n] * h[n]$ by L, neighboring output segments overlap at $P - 1$ points.

Finally, we just add up the output segments using (1) to obtain the output.

Overlap-Add Method



Overlap-Save Method

Basic Idea

We split the input signal $x[n]$ into overlapping segments $x_r[n]$ of length $L+P-1$.

Perform a circular convolution of each input segment $x_r[n]$ with the impulse response $h[n]$, which is of length P using the DFT.

Identify the L-sample portion of each circular convolution that corresponds to a linear convolution, and save it.

This is illustrated below where we have a block of L samples circularly convolved with a P sample filter.

Overlap-Save Method