DFT Based Processing

- Suppose we have a sequence $x[n]$ and another sequence $h[n]$
- assume both starting at $n=0$
- they both have to be the same length (finite) for DFT based processing ⇒ zero padding
- Suppose we execute the following sequence of operations:

(i-a): $x[n] \underset{N}{\overset{DFT}{\longleftrightarrow}} X_N(k)$; (i-b): $h[n] \underset{N}{\overset{DTF}{\longleftrightarrow}} H_N(k)$

(ii) $Y_N(k) = H_N(k)\, X_N(k)$

(iii) $y_p[n] \underset{N}{\overset{DFT}{\longleftrightarrow}} Y_N(k)$ (N point inverse DFT)

- There are at least 3 ways to compute $y_p[n]$

1. actually compute all the DFT's
2. Compute linear convolution $y[n] = x[n] * h[n]$ and then use time-domain aliasing formula

$$y_p[n] = \sum_{\ell=-\infty}^{\infty} y[n-\ell N] \left\{ u[n] - u[n-N] \right\}$$

3. Compute circular convolution of $x[n]$ and $h[n]$.
   See Sect. 7.2.2 and Example 7.2.1 with Fig. 7.2.2 for Graphical Illustration

- An alternative way to compute circular convolution: use of circulant matrix.
    - recall rules of matrix multiplication

- let's illustrate using textbook example 7.2.1

$$x[n] = \{\underset{\uparrow}{2}, 1, 2, 1\} \qquad h[n] = \{\underset{\uparrow}{1}, 2, 3, 4\}$$

- the circular convolution will be of length 4
    ⇒ put as entries of 4x1 vector

$$\underline{y}_p = \begin{bmatrix} y_p[0] \\ y_p[1] \\ y_p[2] \\ y_p[3] \end{bmatrix} \Rightarrow \underset{4x1}{\underline{y}_p} = \underset{4x4}{\underline{H}} \; \underset{4x1}{\underline{x}} \quad \text{where: } \underline{x} = \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix}$$

$\underline{H}$ is a cirulant matrix:

- 1st row is circular time-reverse of $h[n]$

$$\underline{y}_p = \begin{bmatrix} y_p[0] \\ y_p[1] \\ y_p[2] \\ y_p[3] \end{bmatrix} = \begin{bmatrix} 1 & 4 & 3 & 2 \\ 2 & 1 & 4 & 3 \\ 3 & 2 & 1 & 4 \\ 4 & 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 2 \\ 1 \end{bmatrix}$$

- each row is a circular shift by 1 of row above it (wrap around)
- Note, turns out that each column is a circular shift by 1 of the previous column
- Eventually, you can write $\underline{h}$ as the 1st column of $\underline{H}$ in natural order and circularly shift to form the other columns

- multiplying $\underline{H}$ by $\underline{x}$, we obtain same answer as book

$$y_p[n] = \{14, 16, 14, 16\}$$

- I find this to be an easier approach to computing circular convolution relative to writing numbers around a circle as in Fig. 7.2.2

- Check on answer:

$$y[n] = x[n] * h[n] = \{\underset{\uparrow}{2}, 5, 10, 16, 12, 11, 4\}$$

- linear convolution is of length 4+4-1=7
- thus, 7-4=3 nos. at end aliased into 3 numbers at the beginning

- $y_p[n] = y[n] + y[n+4]$, $n = 0, 1, 2, 3$

$$\begin{array}{rl}
y[n]: & \quad\quad\quad\quad\quad\; \{2, 5, 10, 16, 12, 11, 4\} \\
y[n+4]: & \{2, 5, 10, 16, 12, 11, 4\} \\
\hline
y_p[n] = & \quad\quad\quad\quad\quad\; \{14, 16, 14, \underset{\uparrow}{16}\}
\end{array}$$

only unaliased entry

- See Prob. 1 of Exam 3 from Fall 2007 for an example.