Discrete Fourier transform (DFT)

- DFT is used in approximating the spectrum for a finite length sequence.
- DTFT is good for theoretical spectrum analysis of digital signal but not good for computer-aided analysis.
- DFT provides practical approach to numerical computation of DTFT for a finite length sequence.
- Also, we may split an input signal in sets of *N* samples and do analysis for each set (segment or frame) separately.

$$X(\Omega) = \sum_{-\infty}^{+\infty} x[n]\exp[-j\Omega n] = \sum_{n=0}^{N-1} x[n]\exp[-j\Omega n]$$

DFT continued

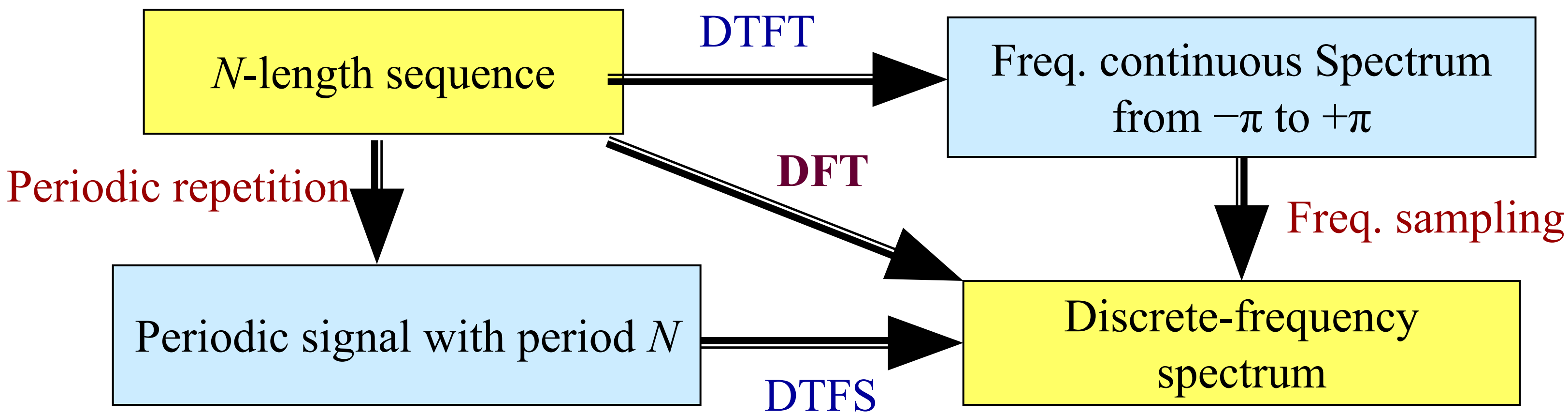
- Approximation to $X(\Omega)$ is obtained by taking frequency samples of $X(\Omega)$ at N equally spaced frequencies $\Omega_k = 2\pi k / N$, $0 \le k \le N - 1$.

$$X[k] = X(\Omega_k) = \sum_{n=0}^{N-1} x[n]\, exp\left(-j\frac{2\pi}{N}nk\right)$$

- Check that the DFT expression is somewhat same as the DTFS expression.
- That means, we do analysis in the DTFS manner.
- That is, for a given set of N samples we calculate the DTFT as if the signal is periodic with these N samples.
- Therefore, one-dimensional DFT and inverse DFT expressions are same as DTFS expressions given

Sampling in frequency

- Discuss frequency sampling in the light of what we have studied in case of time sampling at Nyquist rate.



- For better approximation we may go for *M*-point DFT (*M* > *N*), as if taking an *M*-length sequence created by zero padding to the given *N*-length sequence – discuss in the light of time sampling at a rate greater than Nyquist rate.

Signal recovery from DFT

- Given a non-periodic sequence $x[n]$ of whatever length.
- We take DTFT and then do frequency sampling at N-points to get DFT coefficients.

$$x[n] \;\; \text{– DTFT} \rightarrow \;\; X(\Omega) \;\; \text{–} \; N\text{-point freq. sampling} \rightarrow \;\; X[k]$$

$$X[k] = X(\Omega = 2k\pi/N)$$

- Now we take inverse DFT (expression similar to inverse DTFS) to get back the original signal.
- But, we may or may not get back $x[n]$.

$$X[k] \;\; \text{– IDFT} \rightarrow \;\; \tilde{x}[n]$$

Signal recovery from DFT

- $$\tilde{x}[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\exp\left(j\frac{2\pi}{N}nk\right) \quad n = 0,1,\dots,N-1$$

$$= \frac{1}{N}\sum_{k=0}^{N-1}\left(\sum_{l=-\infty}^{+\infty} x[l]\exp\left(-j\frac{2\pi}{N}lk\right)\right)\exp\left(j\frac{2\pi}{N}nk\right)$$

Signal recovery from DFT

- $$= \sum_{l=-\infty}^{+\infty} x[l] \left(\frac{1}{N} \sum_{k=0}^{N-1} \exp\left(j \frac{2\pi}{N} (n-l)nk \right) \right)$$

$$= \sum_{l=-\infty}^{+\infty} x[l] \ \left(\sum_{m=-\infty}^{+\infty} \delta[(n-l) - mN] \right)$$

$$\tilde{x}[n] \ = \sum_{m=-\infty}^{+\infty} x[n-mN]$$

Signal recovery from DFT

- $x[0], x[1], \ldots, x[N-1], x[N], x[N+1] \ldots, x[2N-1], \ldots$

$$\text{DTFT sampling} \Downarrow \text{and IDFT}$$

$$\tilde{x}[n] = (x[0] + x[N] + \ldots), (x[1] + x[N+1] + \ldots),$$
$$\ldots, (x[N-1] + x[2N-1] + \ldots), \ldots$$

(i.e., every N^{th} sample in $x[n]$ are added up)

Signal recovery from DFT

- Also, check that $\tilde{x}[n] = \tilde{x}[n+kN]$ for any integer k.

$$\tilde{x}[n+kN] = \sum_{m=-\infty}^{+\infty} x[n+kN-mN] = \sum_{m=-\infty}^{+\infty} x[n-(m-k)N]$$

$$\equiv \sum_{p=-\infty}^{+\infty} x[n-pN] = \tilde{x}[n]$$

- That means, $\tilde{x}[n]$ is a periodic sequence with period N.
- Since IDFT is somewhat same as inverse DTFS, this is quite expected.

Signal recovery from DFT

- So, how to determine $\tilde{x}[n]$ from $x[n]$ directly?
  - Split the sequence $x[n]$ into small segments of length N each.
  - Add all the segments to get an N-length sequence.
  - $\tilde{x}[n]$ is the periodic repetition of this N-length sequence.
  - However, we only write down the values of these $\tilde{x}[n]$ for $n = 0$ to $N-1$ as it is unnecessary to repeat the same.

Signal recovery from DFT (contd.)

- For $x[n]$ of length equal to or less than N, $x[n]$ can be recovered from $\tilde{x}[n]$.
  - If length of $x[n]$ is N, then one-period of $\tilde{x}[n]$ = $x[n]$

    Example: $x[n] = \{2, 3, 8, 1\}$ where $N = 4$

    Essentially, $x[n] = \{\ldots, 0, 0, 0, \mathbf{2}, 3, 8, 1, 0, 0, 0, 0, \ldots\}$, where red colored sample corresponds to $x[0]$.

    So, one-period of $\tilde{x}[n] = \{2, 3, 8, 1\}$

  - If length of $x[n]$ is less than N (say $L < N$), then one-period of $\tilde{x}[n]$ = $x[n]$ padded with $(N - L)$ zeros.

    Example: $x[n] = \{2, 3, 8\}$ while $N = 4$

Signal recovery from DFT (contd.)

- Essentially, $x[n] = \{\ldots, 0, 0, 0, \mathbf{2}, 3, 8, 0, 0, 0, 0, 0, \ldots\}$, where red colored sample corresponds to $x[0]$.

  So, one-period of $\tilde{x}[n] = \{2, 3, 8, 0\}$

- If length of $x[n]$ greater than N, there is time-domain aliasing and $x[n]$ cannot be recovered from $\tilde{x}[n]$.

  Example: $x[n] = \{2, 3, 8, 1, 4, 3\}$ while $N = 4$

  Essentially, $x[n] = \{\ldots, 0, 0, 0, \mathbf{2}, 3, 8, 1, 4, 6, 0, 0, \ldots\}$, where red colored sample corresponds to $x[0]$.

  So, one-period of $\tilde{x}[n] = \{2, 3, 8, 1\} + \{4, 6, 0, 0\}$

  $= \{\mathbf{6}, \mathbf{9}, 8, 0\}$

Correspondence to Sampling Theorem

- Continuous-time periodic signal in time domain $\rightleftharpoons$ discrete-frequency non-periodic spectrum (FS coeff's)
- In the same way, continuous-frequency periodic spectrum $X(\Omega)$ in frequency domain $\rightleftharpoons$ discrete-time non-periodic signal $x[n]$.
- Time domain signal sampled with time-interval T_s $\rightleftharpoons$ frequency domain spectrum gets repeated after every $2\pi / T_s = 2\pi f_s = \omega_s$ rad / sec
- Similarly, frequency domain spectrum $X(\Omega)$ sampled at N points with frequency-interval $2\pi/N$ rad. $\rightleftharpoons$ time domain signal gets repeated after every $2\pi/\left(\frac{2\pi}{N}\right)$ = N samples.

Correspondence to Sampling Theorem

- According to sampling theorem, $\omega_s \geq$ total band-width of spectrum in rad/sec (incl. both pos. and neg. sides of the frequency band) is required for exact reconstruction of the original signal from the time-domain samples.
- Similarly, here we require $N \geq$ total length of signal for exact recovery of the original signal from the frequency-domain samples (DFT coefficients), as checked before.
- For $\omega_s >$ spectrum band-width (in rad/sec), we get a guard-band in between the repeated spectrum repetition.
- Similarly, for $N >$ signal length L, we get a guard-time in between the sequence $x[n]$ repetition given by the zero padding of length $N-L$, as mentioned before.

Correspondence to Sampling Theorem

- That is why, for an *N*-length sequence we generally do *N*-point DFT / IDFT.
- However, we may go for *N*-point DFT even for an *L*-length sequence ($L < N$). The same has been said before that for an *N*-length sequence, we may go for *M*-point DFT ($M > N$) for better approximation of the spectrum.
- We will discuss soon that this more-point DFT / IDFT is also required for carrying out convolution via multiplication in frequency domain – multiplying DFT coefficients and then taking IDFT of these DFT coefficient products.

Correspondence to Sampling Theorem

- For ω_s < spectrum band-width (in rad/sec), we get aliasing due to spectrum overlap.
- In a similar fashion, for N < signal length L, we get overlapping of the sequences where the last $L-N$ samples in the sequence $x[n]$ gets added with the first $L-N$ samples of the sequence. This is aliasing in time, as discussed earlier with an example.
- Another example: $x[n]$ = {2, 3, 8, 1, 4, 3, 5, 7, 2} and we do 4-point DFT followed by 4-point IDFT (i.e. N = 4)

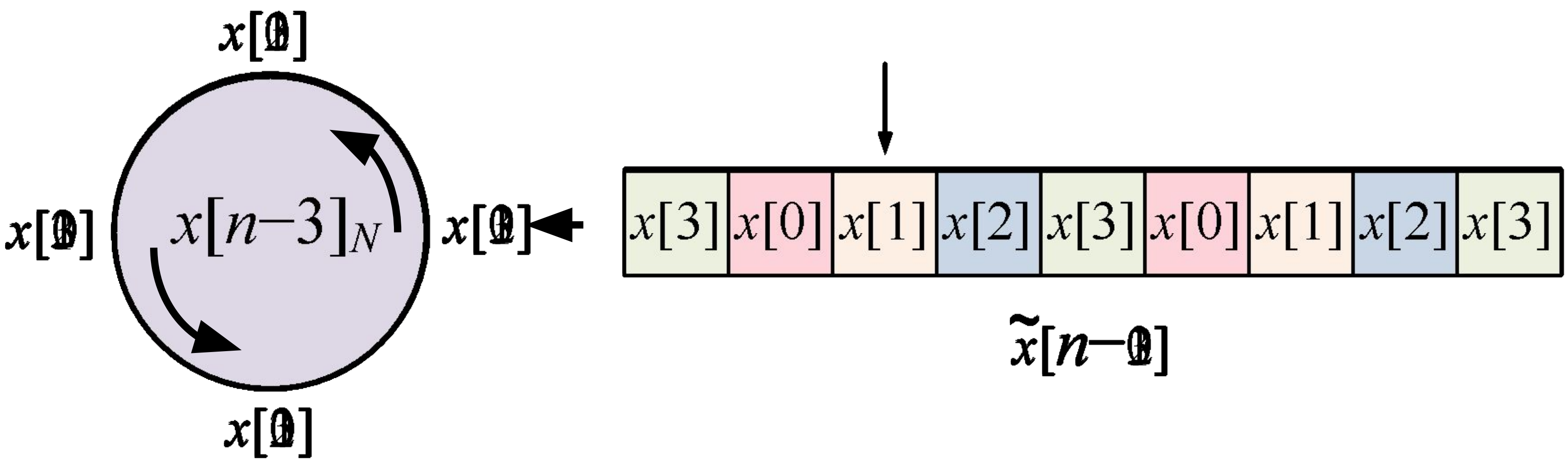
  Then, one-period of the IDFT result $\tilde{x}[n]$ = {8, 6, 13, 8}

DFT properties

- **Linearity:** $ax[n] + by[n] \rightleftharpoons aX[k] + bY[k]$
- **Circular shift:** $x[n - n_0]_N \rightleftharpoons \exp\left[-j\frac{2\pi}{N}kn_0\right]X[k]$
- $x[n - n_0]_N$ denotes $x[n]$ circularly shifted by n_0 samples.
- What do you mean by circular shift?
  - $x[n]$ is finite-length sequence of length *N*.
  - But, DFT is essentially DTFS of the periodic repetition of $x[n]$ which is $\tilde{x}[n]$.
  - That is, we can say $\tilde{x}[n - n_0] \rightleftharpoons \exp\left[-j\frac{2\pi}{N}kn_0\right]X[k]$ when $\tilde{x}[n]$ linearly shifted by n_0 samples towards right.

DFT properties

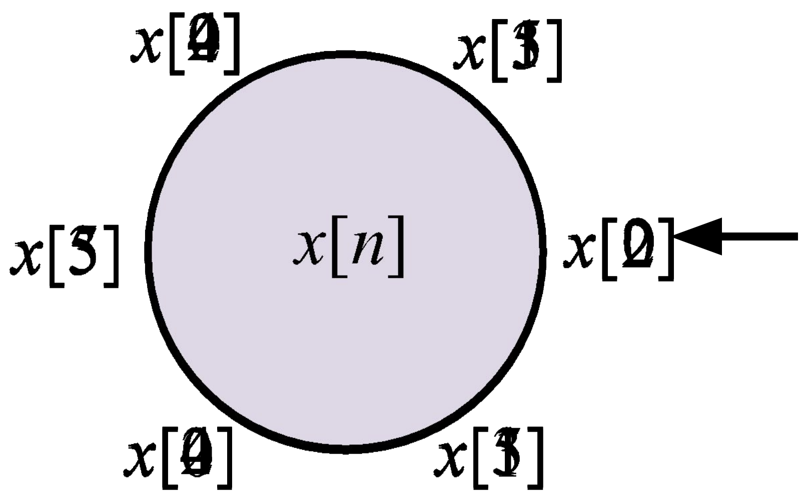
- How can we express $\tilde{x}[n-n_0]$ in terms of $x[n]$?
- Write the N samples of $x[n]$ on a circle; our convention is to write in anti-clockwise direction for increasing n.
- Shift (rotate) the samples in the same anti-clockwise direction corresponding to linear right-shift of $\tilde{x}[n]$.



DFT properties

- What is $x[n_0 - n]_N$?
  - $x[n_0 - n]_N = x[-n + n_0]_N \Rightarrow$ First circularly shift $x[n]$ clockwise by n_0 positions, followed by scaling by -1, i.e. reverse the order about the origin on the circle.
  - See for N = 6 and n_0 = 2 case
- What is $x[N - n]_N$?
  - $x[N - n]_N$

    $= \{x[0], x[N-1], \dots, x[1]\}$



DFT properties

- **Symmetry:**
  - For real *x*[*n*]: $X[k] \;=\; X^*[-k]$ (conjugate symmetry)

    But DFT defined only for *k* = 0 to *N*−1

$$X[-k] = \sum_{n=0}^{N-1} x[n] \exp\left(-j\frac{2\pi}{N}n(-k)\right)$$
$$= \sum_{n=0}^{N-1} x[n] \exp\left(-j\frac{2\pi}{N}n(N-k)\right) \;=\; X[N-k]$$

Therefore, $X^*[-k] \;=\; X^*[N-k]$

DFT properties

- $$X^*[-k] \begin{cases} = X^*[N] \equiv X^*[0] \text{ for } k = 0 \\ = X^*[N-k] \text{ for } k = 1 \text{ to } N-1 \end{cases}$$

  That is, $X^*[-k]$

  $$= \{X^*[0],\ X^*[N-1], \ldots, X^*[1]\} \equiv X^*[N-k]_N$$

  - For imaginary $x[n]$: $X[k] = -X^*[-k] = -X^*[N-k]_N$

    (conjugate anti-symmetry)

Convolution

- $y[n] = x[n] * h[n]$
- **Case 1:** Both $x[n]$ and $h[n]$ are periodic
  - with same period N.
  - With different periods N_1 and N_2
- **Case 2:** $x[n]$ periodic with period N but $h[n]$ non-periodic of length L.
  - $N \geq L$
  - $N < L$
- **Case 3:** Both $x[n]$ and $h[n]$ are non-periodic of lengths N_1 and N_2

Circular Convolution

- **Both $x[n]$ and $h[n]$ are periodic with same period N –**
- In general, $y[n] = x[n] * h[n] = \sum_{l=-\infty}^{+\infty} x[l]h[n-l]$
  - Reflect $h[l]$ to get $h[-l]$ → multiply and add to get $y[0]$
  - Shift $h[-l]$ by n samples towards right for positive n and towards left for negative n to get $h[n-l]$ → multiply and add to get $y[n]$.

$x[l]$

...	$x[0]$	$x[1]$	$x[2]$	$x[3]$	$x[0]$	$x[1]$	$x[2]$	$x[3]$	$x[0]$	...
...	$h[0]$	$h[3]$	$h[2]$	$h[1]$	$h[0]$	$h[3]$	$h[2]$	$h[1]$	$h[0]$	...

Circular Convolution

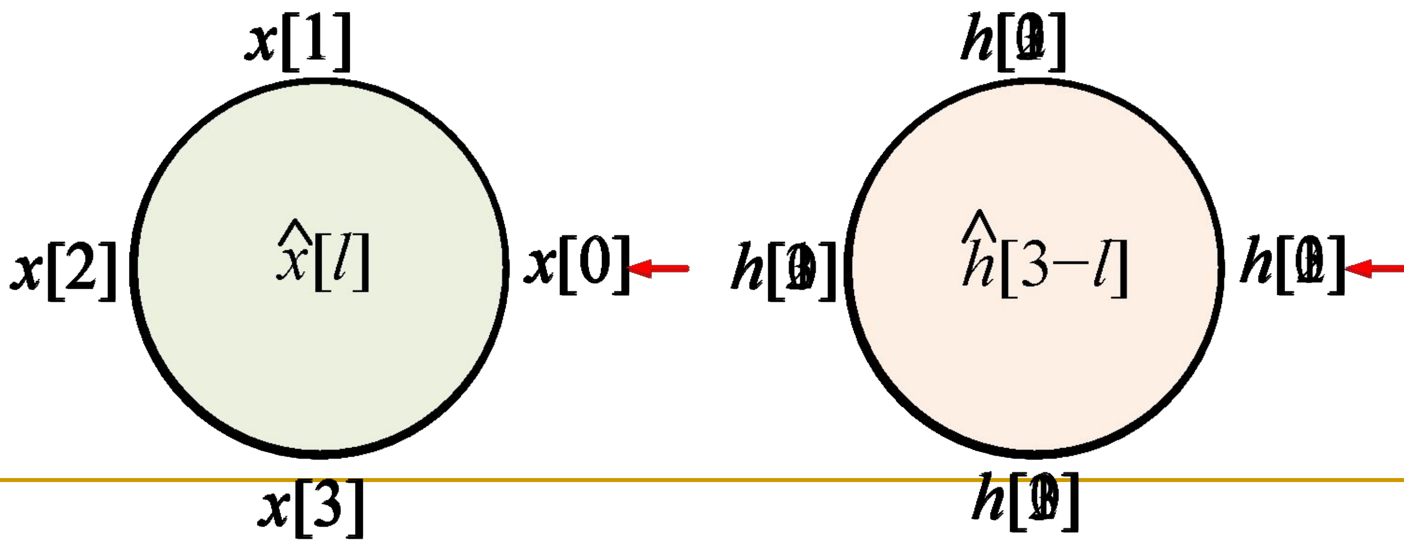
- So, we observe two facts:
  - Taking summation over $l = -\infty$ to $+\infty$ will give each $y[n]$ infinite valued since both $x[n]$ and $h[n]$ are of infinite duration (system with periodic $h[n]$ is unstable).
  - Output $y[n]$ is also periodic with same period *N*.
- Therefore, in this case we calculate convolution over one period only to get one period of $y[n]$, denoting as $\hat{y}[n]$.

$$\hat{y}[n] = \sum_{l=0}^{N-1} x[l]h[n-l]$$

- Now, since periodic, this can be done more conveniently in a circular arrangement rather than linear convolution.

Circular Convolution

- Arrange $\hat{x}[l]$ in anti-clockwise, and then arrange $\hat{h}[l]$ clockwise to get $\hat{h}[-l]$; $\hat{x}[l]$ and $\hat{h}[l]$ denote only one period of $x[n]$ and $h[n]$ and so of finite length *N* each.
- Multiply point-to-point and add to get $\hat{y}[0]$.
- Shift $h[-l]$ by *n* samples anti-clockwise, *n* = 1 to *N*−1, to get $\hat{h}[n-l]$. Multiply and add to get $\hat{y}[n]$.



Circular Convolution

- This way of doing convolution is called *N*-point circular convolution, denoted as

$$\hat{y}[n] = \hat{x}[n] \circledast_N \hat{h}[n]$$

  while the general convolution may be specifically called linear convolution or simply convolution.

- As we understand now, circular convolution is applied in convolving two periodic signals, but taking only one period of each.

- Now, how to do the same in the frequency domain?

Circular Convolution

- Recall DTFS property:

$$\hat{x}[n] \circledast_N \hat{h}[n] = \sum_{l=0}^{N-1} x[l]h[n-l] \rightleftharpoons N\, X_{DTFS}[k] \times H_{DTFS}[k]$$

$$\Rightarrow \quad \hat{y}[n] = \hat{x}[n] \circledast_N \hat{h}[n] = \text{IDTFS}\, \{N\, X_{DTFS}[k] \times H_{DTFS}[k]\}$$

- Now, for the same input sequence DFT coefficients = $N \times$ DTFS coefficients, i.e., $X[k] = N \times X_{DTFS}[k]$
- And, for same input coefficients set IDTFS = $N \times$ IDFT
- So, we may also find $\hat{y}[n]$ from DFTs of $\hat{x}[n]$ and $\hat{h}[n]$.

Circular Convolution

- $\hat{y}[n] = \hat{x}[n] \circledast_N \hat{h}[n]$ = IDTFS $\{N\, X_{DTFS}[k] \times H_{DTFS}[k]\}$

$$= N \times \text{IDFT}\ \{N\, X_{DTFS}[k] \times H_{DTFS}[k]\}$$

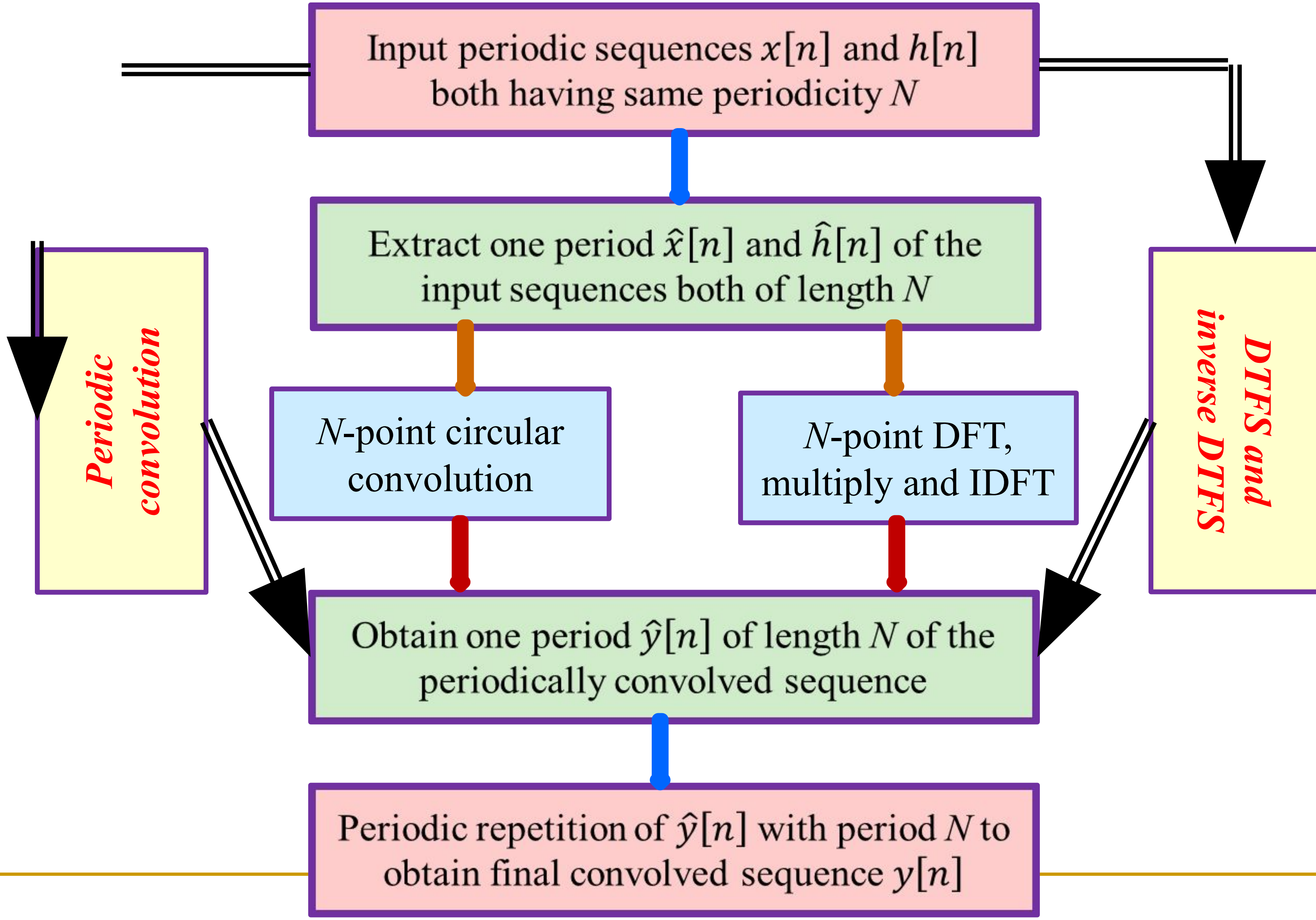
$$= \text{IDFT}\ \{N\, X_{DTFS}[k] \times N\, H_{DTFS}[k]\}$$

$$= \text{IDFT}\ \{N\, X_{DTFS}[k] \times N\, H_{DTFS}[k]\} = \text{IDFT}\ \left\{\hat{X}[k] \times \hat{H}[k]\right\}$$

$\Rightarrow$ $\hat{Y}[k] = \hat{X}[k] \times \hat{H}[k]$ are the DFT coefficients of $\hat{y}[n]$.

- Therefore, to perform periodic convolution of two periodic sequences of same period *N*, take *N*-point DFTs of one period of each, multiply the DFT coefficients and then take *N*-point IDFT to get *N*-length $\hat{y}[n]$.

Periodic Convolution



Convolution Property

- **Convolution property of DFT** that stems out from here

$$\hat{x}[n] \circledast_N \hat{h}[n] \quad \rightleftharpoons \quad \hat{X}[k] \times \hat{H}[k] \text{ (N-point DFT)}$$

for any finite-length sequences $\hat{x}[n]$ and $\hat{h}[n]$.

- So, summarizing the convolutional properties:
  - FS: $T\,X[k]H[k] \rightleftharpoons$ one period $x(t) * h(t)$, T = period
  - FT: $X(\omega)H(\omega) \rightleftharpoons x(t) * h(t)$
  - DTFT: $X(\Omega)H(\Omega) \rightleftharpoons x[n] * h[n]$
  - DTFS: $N\,X[k]H[k] \rightleftharpoons$ one period $x[n] * h[n]$, N = period
  - DFT: $X[k]H[k] \rightleftharpoons x[n] \circledast_N h[n]$

Convolution Property

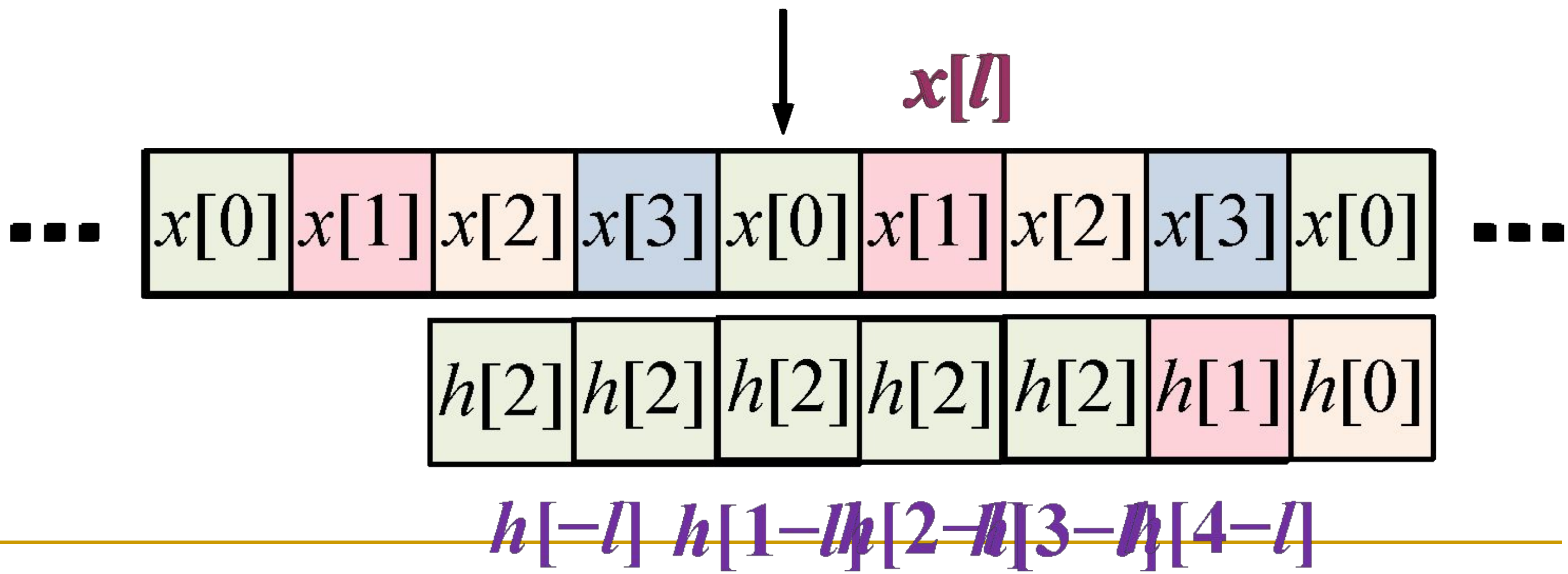
- Taking *N*-point DFT of finite-length sequences $x[n]$ and $h[n]$ followed by multiplication of the DFT coefficients and then taking IDFT essentially converts the sequences as if they are both periodic with the same period *N* and then we do periodic convolution of them.
- For computing *N*-point circular convolution of two sequences of whatever lengths via DFT / IDFT, we have to take same *N*-point DFT for both sequences since we do point-to-point multiplication of DFT coefficients.
  - The value of *N* needs to be chosen judiciously as we have seen earlier that there is a possibility of time-aliasing after we take IDFT.

Circular Convolution contd.

- Observe that for doing circular convolution, both the sequences must be of same length since we are doing point-to-point multiplication (this is not a necessary condition in linear convolution).
  - Both sequences must be of length N to apply N-point circular convolution on them.
  - If not we have to do something appropriate to make them equal length sequences.
  - In the case discussed we have taken both sequences of same periodicity and so there was no problem.
  - Now we will take up other cases.

Circular Convolution contd.

- **Both $x[n]$ and $h[n]$ are periodic but with different periods N_1 and N_2 –**
  - Take N = LCM (N_1, N_2)
- **$x[n]$ periodic with period N and $h[n]$ non-periodic with length L, where $N \geq L$ –**

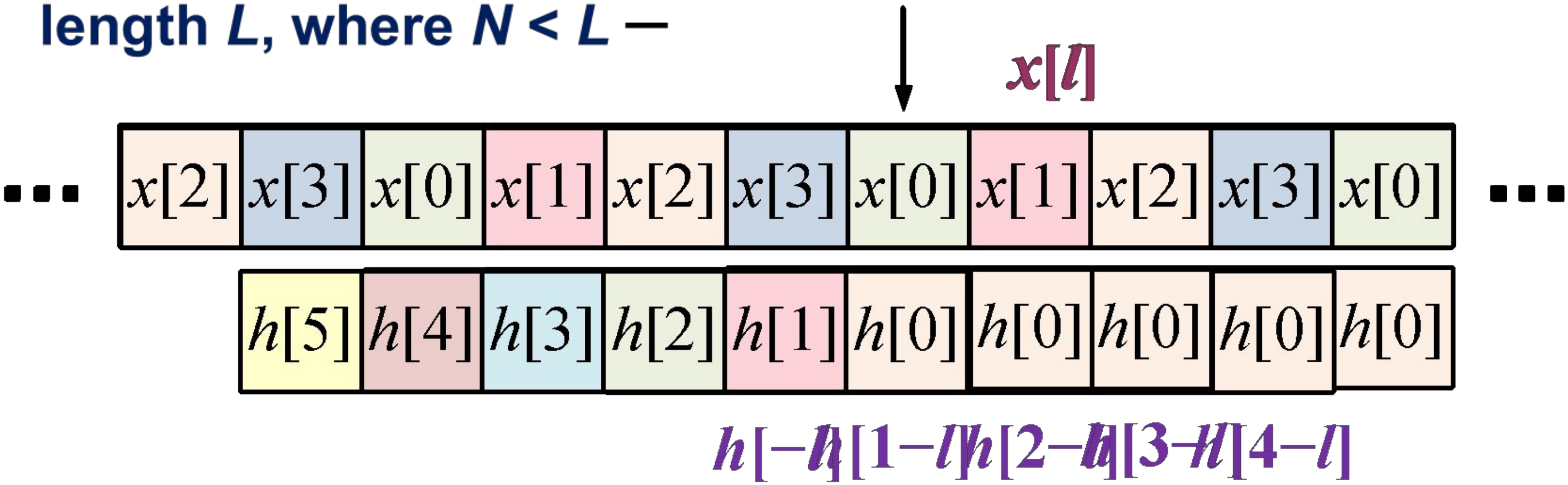


Circular Convolution contd.

- Here again $y[n]$ is periodic with period *N.*
- But, unlike the case of both $x[n]$ and $h[n]$ periodic sequences, here each $y[n]$ is finite-valued (system with finite-length impulse response $h[n]$ is stable)
- One period of $y[n]$ can be computed by doing *N*-point circular convolution; for $N > L$, length of $h[n]$ extended by zero-padding.
- Alternatively, perform *N*-point DFT / IDFT.
  - Essentially gives periodic convolution of $x[n]$ with periodic repetition of $h[n]$ padded with $N - L$ zeros.
  - Since $N \geq L$ there is no question of time-aliasing.

Circular Convolution contd.

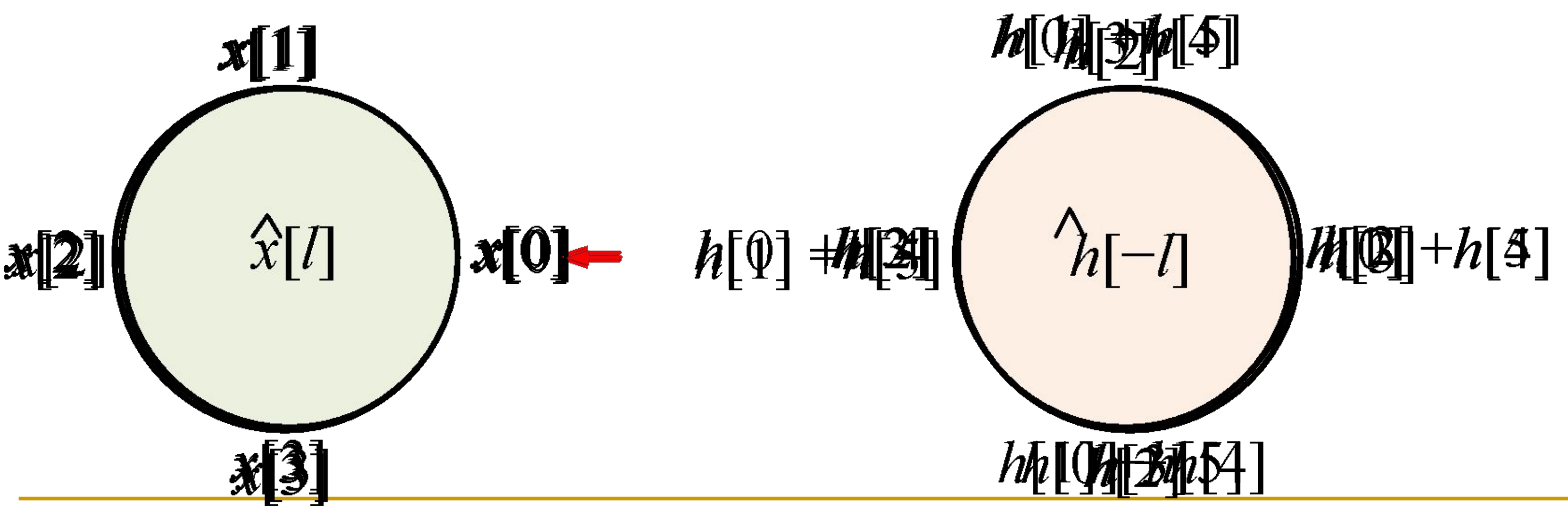
- $x[n]$ periodic with period N and $h[n]$ non-periodic with length L, where $N < L$ –



$$\hat{y}[0] = x[0](h[0] + h[4]) + x[3](h[1] + h[5]) + x[2]h[2] + x[1]h[3]$$
$$\hat{y}[1] = x[3](h[0] + h[4]) + x[2](h[1] + h[5]) + x[1]h[2] + x[0]h[3]$$
$$\hat{y}[2] = x[2](h[0] + h[4]) + x[1](h[1] + h[5]) + x[0]h[2] + x[3]h[3]$$
$$\hat{y}[3] = x[1](h[0] + h[4]) + x[0](h[1] + h[5]) + x[3]h[2] + x[2]h[3]$$

Circular Convolution contd.

- So, here also one period of $y[n]$ can be computed by doing *N*-point circular convolution. Since, length of $h[n]$ is more than *N*, $h[n]$ is time-aliased as follows.



Circular Convolution contd.

- Alternatively, perform *N*-point DFT / IDFT.
  - Taking N-point DFT of finite-length $h[n]$ may be thought of as if $h[n]$ is periodically repeated with period *N.*
  - But, since length *L* of $h[n]$ is more than the number of DFT points *N*, there is time-aliasing in the periodically repeated $h[n]$ sequence. This is desired since we want circular convolution of $x[n]$ with time-aliased $h[n]$ as seen above.
  - Thus, DFT / IDFT approach essentially gives periodic convolution of $x[n]$ with periodic repetition of time-aliased $h[n]$ with period *N*, as desired.

Circular Convolution contd.

- **Both $x[n]$ and $h[n]$ are non-periodic of lengths N_1 and N_2 –**
  - Convolution of two non-periodic signals gives a finite-length non-periodic output; the length of the output is $N_1 + N_2 - 1$.
  - Convolution here is obtained by linear convolution.
  - This can only be computed via frequency domain using DTFT / IDTFT.
- However, our objective is to do the same via DFT / IDFT since DTFT / IDTFT cannot be performed in a digital processor.

Circular Convolution contd.

- But, we have seen that DFT / IDFT gives circular convolution and not linear convolution.
- How is circular convolution related to linear convolution?

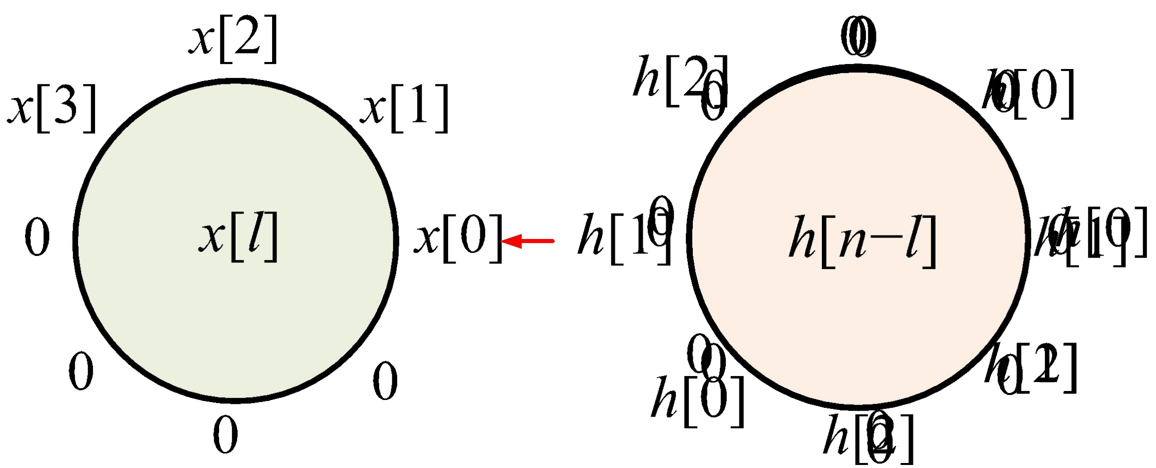
$x[0]$	$x[1]$	$x[2]$	$x[3]$

$h[2]$	$h[2]$	$h[2]$	$h[2]$	$h[2]$	$h[2]$	$h[1]$	$h[0]$

$$y[0] = x[0]h[0]$$
$$y[1] = x[0]h[1] + x[1]h[0]$$
$$y[2] = x[0]h[2] + x[1]h[1] + x[2]h[0]$$
$$y[3] = x[1]h[2] + x[2]h[1] + x[3]h[0]$$
$$y[4] = x[2]h[2] + x[3]h[1]$$
$$y[5] = x[3]h[2]$$

Circular Convolution contd.

- Now, we compare this linear convolution to N-point circular convolution for different values of N.

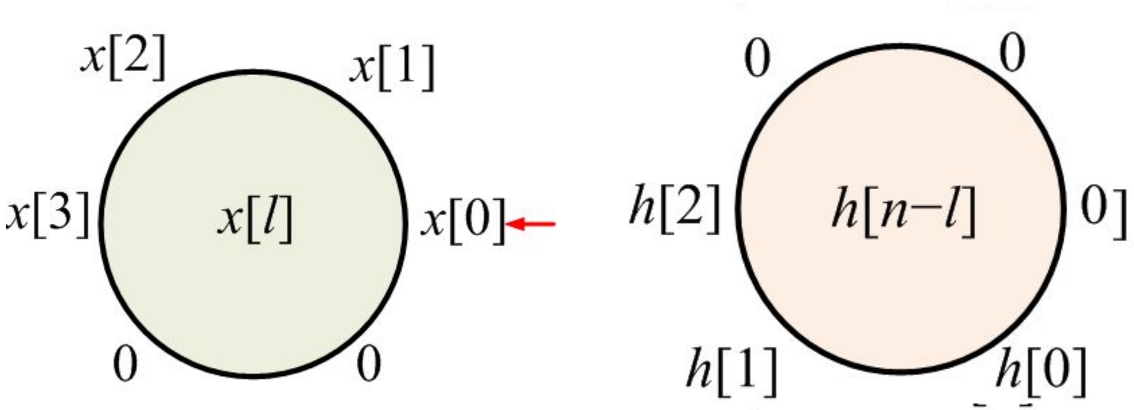


- So, for $N = 8$,
  circular convolution
  = linear convolution

$$
\begin{aligned}
y[0] &= x[0]h[0] \\
y[1] &= x[0]h[1] + x[1]h[0] \\
y[2] &= x[0]h[2] + x[1]h[1] + x[2]h[0] \\
y[3] &= x[1]h[2] + x[2]h[1] + x[3]h[0] \\
y[4] &= x[2]h[2] + x[3]h[1] \\
y[5] &= x[3]h[2]
\end{aligned}
$$

Circular Convolution contd.

- Now, we compare this linear convolution to *N*-point circular convolution for different values of *N*.

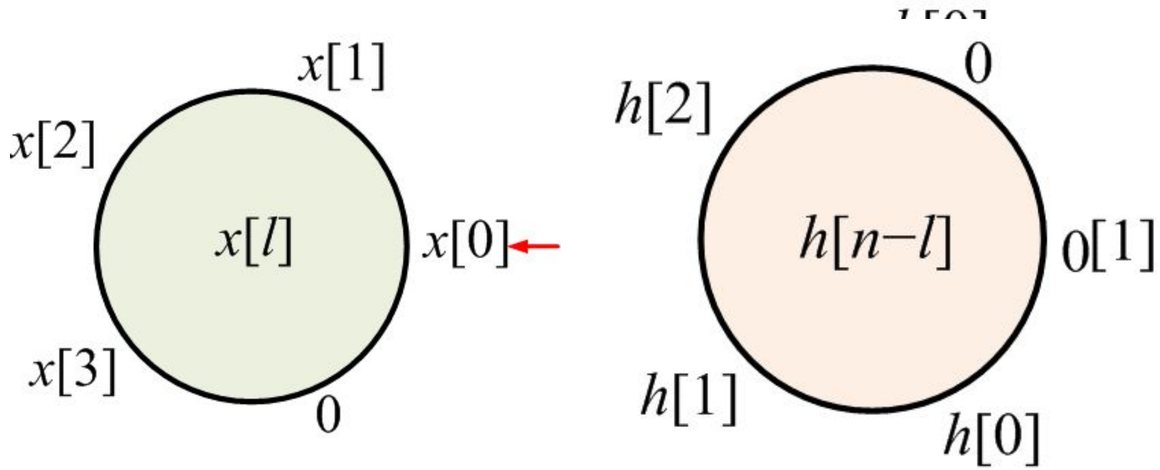


- So, for $N = 6$, circular convolution = linear convolution

$$
\begin{aligned}
y[0] &= x[0]h[0] \\
y[1] &= x[0]h[1] + x[1]h[0] \\
y[2] &= x[0]h[2] + x[1]h[1] + x[2]h[0] \\
y[3] &= x[1]h[2] + x[2]h[1] + x[3]h[0] \\
y[4] &= x[2]h[2] + x[3]h[1] \\
y[5] &= x[3]h[2]
\end{aligned}
$$

Circular Convolution contd.

- Now, we compare this linear convolution to N-point circular convolution for different values of N.



- So, for $N = 5$,

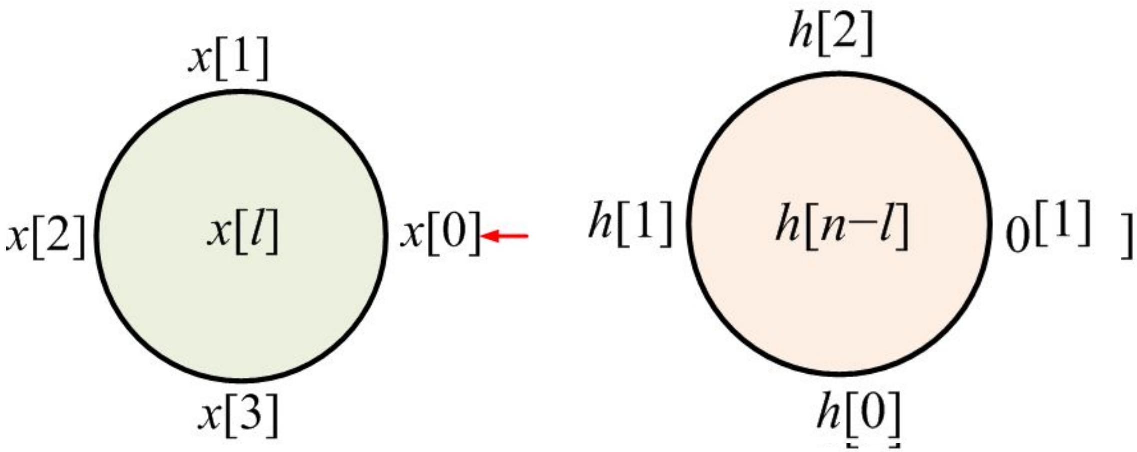
  circular convolution

$$= y[0]+y[5],\ y[1],\ y[2],\ y[3],\ y[4]$$

$$y[0] = x[0]h[0]$$
$$y[1] = x[0]h[1] + x[1]h[0]$$
$$y[2] = x[0]h[2] + x[1]h[1] + x[2]h[0]$$
$$y[3] = x[1]h[2] + x[2]h[1] + x[3]h[0]$$
$$y[4] = x[2]h[2] + x[3]h[1]$$
$$y[5] = x[3]h[2]$$

Circular Convolution contd.

- Now, we compare this linear convolution to N-point circular convolution for different values of N.



- So, for N = 4,

  circular convolution

= $y[0]+y[4]$, $y[1]+y[5]$, $y[2]$, $y[3]$

$$
\begin{aligned}
y[0] &= x[0]h[0] \\
y[1] &= x[0]h[1] + x[1]h[0] \\
y[2] &= x[0]h[2] + x[1]h[1] + x[2]h[0] \\
y[3] &= x[1]h[2] + x[2]h[1] + x[3]h[0] \\
y[4] &= x[2]h[2] + x[3]h[1] \\
y[5] &= x[3]h[2]
\end{aligned}
$$

Circular Convolution contd.

- For $M \neq N$, N-point circular convolution is different from M-point circular convolution.
- Circular convolution can be derived from linear convolution $y[n] = x[n] * h[n]$ as linear convolution + time-aliasing. That is,

$$\hat{x}[n] \circledast_N \hat{h}[n] = \text{one period of } \sum_{m=-\infty}^{+\infty} y[n+mN]$$

  - On the other hand, $y[n]$ can be obtained from circular convolution if there is no time aliasing. And, to avoid time aliasing, $N \geq N_1 + N_2 - 1$.
- Thus, linear convolution of two finite length sequences can be obtained via DFT / IDFT if N is taken as above.

Summarizing

- Both $x[n]$ and $h[n]$ are periodic with same period N (or N = lcm[N_1, N_2]): General linear convolution $x[n] * h[n]$ not valid; compute one period of periodic convolution
  - via N-point circular convolution of $\hat{x}[n]$ and $\hat{h}[n]$.
  - via N-point DFT / IDFT of $\hat{x}[n]$ and $\hat{h}[n]$.
  - via linear convolution followed by time-aliasing – perform linear convolution giving output sequence of length $2N - 1$, then overlap last $N - 1$ output samples with the first $N - 1$ output samples resulting in the required N-length one period of the periodic convolution.

Summarizing

- $x[n]$ periodic with period *N* but $h[n]$ is of finite-length *L*: Compute one period of linear convolution $x[n] * h[n]$
  - via *N*-point circular convolution of $\hat{x}[n]$ and $\hat{h}[n]$; for *L* = *N*, $\hat{h}[n]$ = $h[n]$, for *L* < *N*, $h[n]$ is zero-padded and for L > N, $h[n]$ is time-aliased to make $\hat{h}[n]$ *N*-length long.
  - via *N*-point DFT / IDFT of $\hat{x}[n]$ and $h[n]$.
  - via linear convolution followed by time-aliasing.
  - Note that here taking the convolution sum over $l = -\infty$ to $+\infty$ is as good as taking the sum over finite length *L*. So, here circular convolution gives one period of the actual linear convolution $x[n] * h[n]$.

Summarizing

- Both $x[n]$ and $h[n]$ are non-periodic of lengths N_1 and N_2: Compute Linear convolution $x[n] * h[n]$ computed
  - via linear convolution as usual.
  - via N-point circular convolution of $x[n]$ and $h[n]$, where we take $N \geq N_1 + N_2 - 1$.
  - via N-point DFT / IDFT of $x[n]$ and $h[n]$.