Digital signal processing Lab

Exp. 9: Zooming of an image through circular convolution

Circular or periodic convolution (what we usually DON'T want! But be careful, in case we do want it!)

Remembering that convolution in the TD is multiplication in the FD (and vice-versa) for both continuous and discrete infinite length sequences, we would like to see what happens for periodic, finite-duration sequences.

So let's form the product of the DFS in the FD and see what we get after an IDFS back to the TD.

We have the FT pairs (the relationship is symmetric)

$$\tilde{x}_1(n) * \tilde{x}_2(n) \Leftrightarrow \tilde{X}_1(k)\tilde{X}_2(k)$$
$$\tilde{x}_1(n)\tilde{x}_2(n) \Leftrightarrow \tilde{X}_1(k) * \tilde{X}_2(k)$$

This is fine mathematically.

But now, unless we are careful, what we get on the computer is known as circular convolution.

It comes from the fact that the <u>DFT is periodic</u>, with the period equal to the length of the finite sequence.

Remember that for "regular" convolution - we ***padded the sequences with zeros***, flipped one and slid them along one another, so for the sequences (0, 2, 1, 0) and (0, 3, 4, 0) we had

$$\begin{array}{l} \ldots 0 \;\; 2 \;\; 1 \;\; 0 \ldots \\ \quad \bullet \;\; \bullet \;\; \bullet \\ \ldots 0 \;\; 4 \;\; 3 \;\; 0 \ldots \\ \hline \end{array}$$

$$0 + 0 + 6 + 0 + 0 = 6 = z_0$$

$$\begin{array}{l} \ldots 0 \;\; 2 \;\; 1 \;\; 0 \ldots \\ \quad \bullet \;\; \bullet \;\; \bullet \;\; \bullet \\ \ldots 0 \;\; 4 \;\; 3 \;\; 0 \ldots \\ \hline \end{array}$$

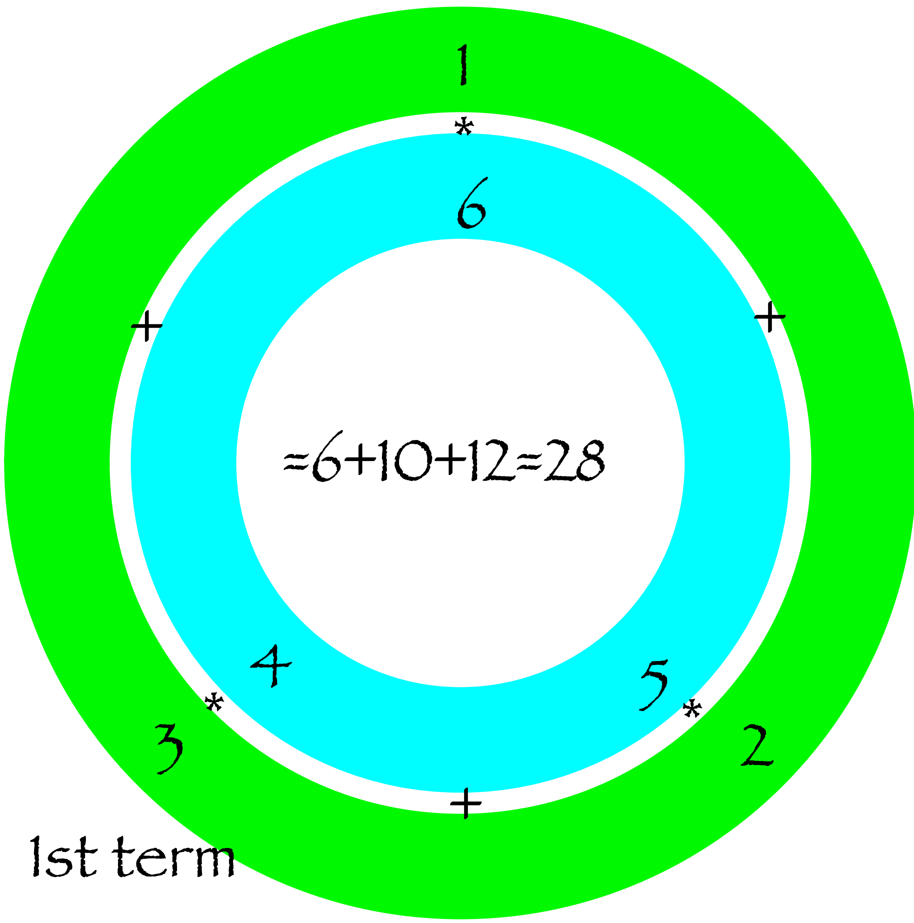
$$0 + 8 + 3 + 0 + 0 = 11 = z_1$$

$$\begin{array}{l} \ldots 0 \;\; 2 \;\; 1 \;\; 0 \ldots \\ \qquad \bullet \;\; \bullet \;\; \bullet \\ \qquad \ldots 0 \;\; 4 \;\; 3 \;\; 0 \ldots \\ \hline \end{array}$$

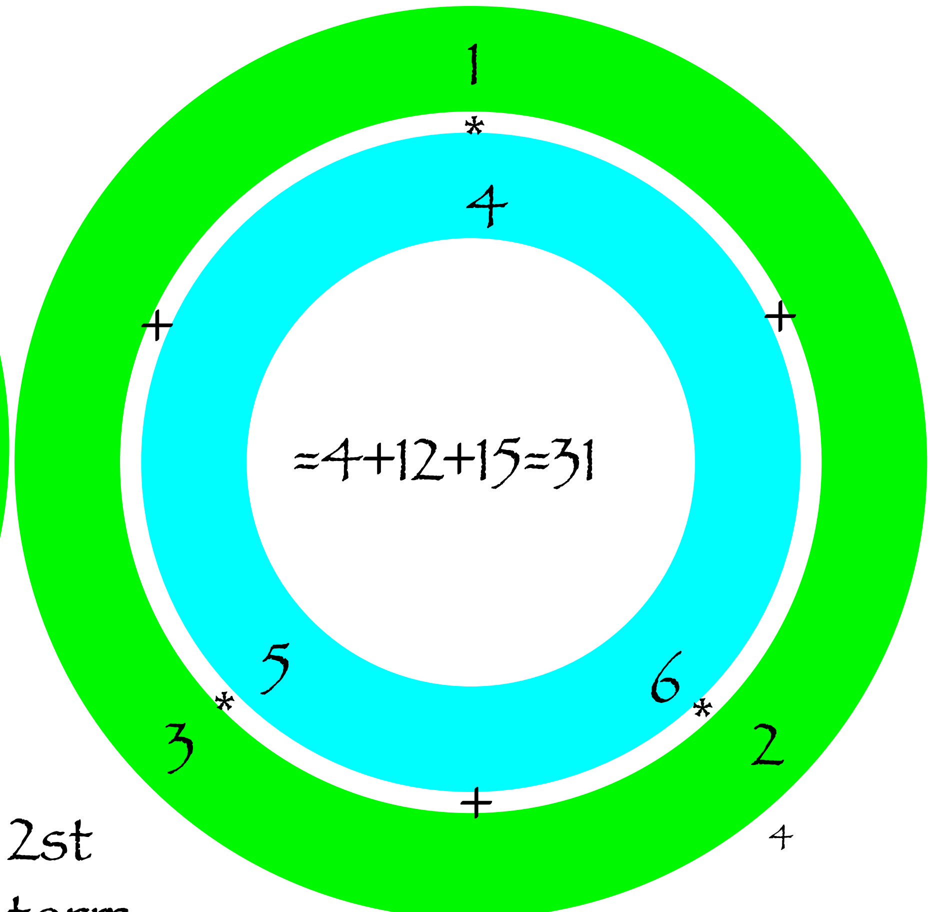
$$0 + 0 + 4 + 0 + 0 = 4 = z_2$$

3

In circular or periodic convolution we can look at the *N* point sequences as being distributed on a circle due to the periodicity. Now we do the same thing (line up, multiply and add, then shift), but with concentric circles. Let's convolve $x_1(n)=(1,2,3)$ and $x_2(n)=(4,5,6)$. One sequence is distributed clockwise and the other counterclockwise and the shift of the inner circle is clockwise.



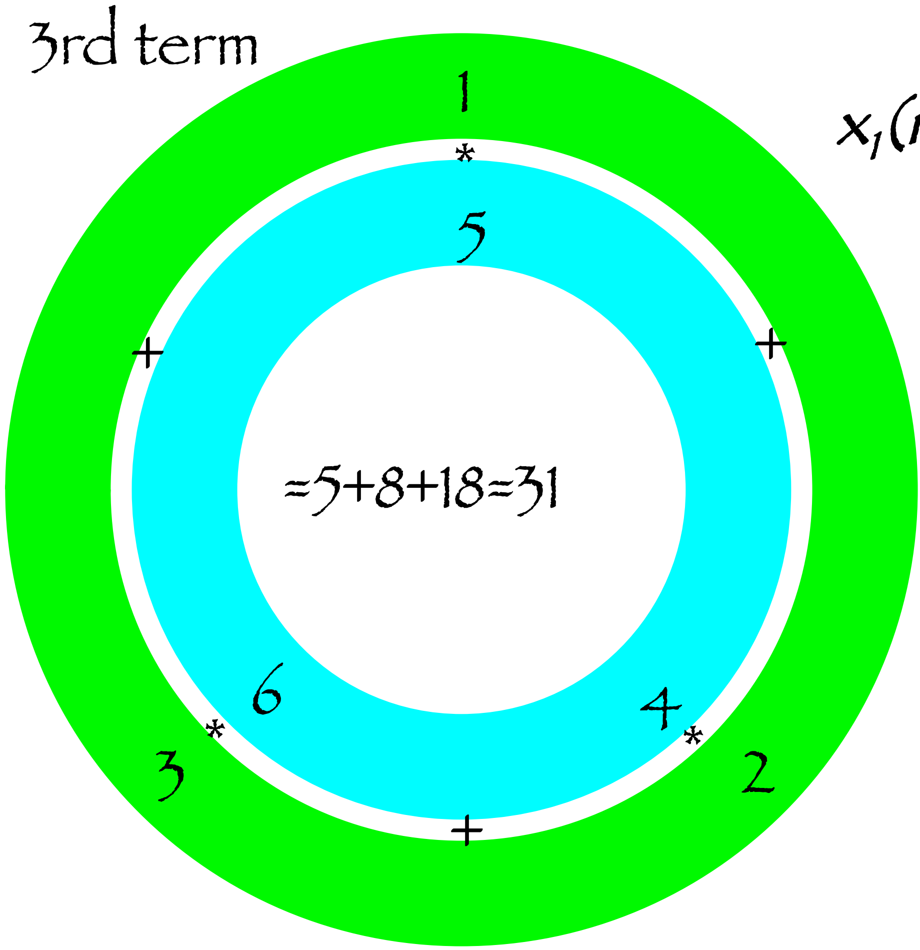
1st term



2st term

4

So $y(n)$ is obtained in a manner reminiscent of convolution with the modifications that $x_1(m)$ and $x_2(m-n)$ are periodic in m with period N (this makes the "circular" part) and consequently so is their "product" (periodic in m with period N and circular). Also remember that the summation is carried out over only <u>ONE period</u>.



$x_1(n) * x_2(n) = (1,2,3) * (4,5,6) = (18,31,31).$

We have the same symmetry as before

$$\tilde{y}(n) = \sum_{k=0}^{N-1} \tilde{x}_1(k)\,\tilde{x}_2(n-k)$$

$$\tilde{y}(n) = \sum_{k=0}^{N-1} \tilde{x}2(k)\,\tilde{x}_1(n-k)$$

5

Compared to our linear convolution "machine" we make ONE of the sequences periodic.

… 0 1 2 3 0 …

• • • •

…6 5 4 6 5 4 6 5 …

0+ 0+ 6+ 10+12 = 28 = z_0

… 0 1 2 3 0 …

• • • •

…6 5 4 6 5 4 6 5 …

0+ 0+ 4+12+15 = 31 = z_1

… 0 1 2 3 0 …

• • • •

…6 5 4 6 5 4 6 5 …

0+ 5+ 8+18+ 0 = 31 = z_2

Review of Circular Convolution

- For DFT, time domain **circular convolution** implies frequency domain multiplication, and vice versa.

- Circular Convolution of two N-point sequences:

$$x_3[n] = x_1[n] \otimes x_2[n] = x_2[n] \otimes x_1[n]$$

$$\equiv \sum_{m=0}^{N-1} x_2[m] x_1[((n-m))_N] = \underbrace{\sum_{m=0}^{N-1} x_2[m] x_1[(n-m) \bmod N]}_{\text{circular convolution}}$$

- Symbol for representing circular convolution: $\otimes$ or Ⓝ.

Speeding up Circular Convolution Using FFT

- **Circular convolution can be computed more efficiently by using FFT.**
- Direct circular convolution of two lengh-N sequences: takes the time complexity of $O(N^2)$.
- Circular convolution by FFT and inverse FFT:
  - Apply FFT to the length-N signal (transform to the frequency domain by FFT)
  - Perform multiplication in the frequency domain, and then apply inverse FFT (inverse transform to the time domain)
- Since both FFT and inverse FFT take $O(Nlog(N))$ and multiplication takes $O(N)$, the time complexity is $O(Nlog(N))$.

Linear Convolution Using FFT

- However, **linear convolution cannot be speeded up directly by FFT**, since multiplication in the frequency domain of DFT (FFT) implies circular convolution but not linear convolution.

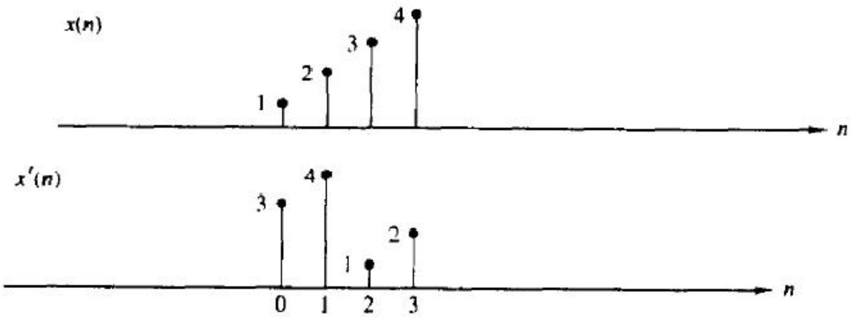
- How can we speed up linear convolution by FFT?

- **Trick**: Seeking the relationships between linear convolution and circular convolution.

- A useful property
  - The linear convolution of two finite-length sequences (with lengths being L and P respectively) is equivalent to circular convolution of the two N-point ($N \geq L+P-1$) sequences obtained by zero padding.

Circular Shift

Doing normal shift on $x_p(n)$ is equivalent to do circular shift on $x(n)$



In previous example, the samples from $x_p(n-2)$ 0 to N - 1 result in a circular shifted version of $x(n)$ by 2.

Circular Shift

This can be expressed by using the following formula

$$x'(n) = x(n-k, \text{mod ule } N)$$
$$\equiv x((n-k))_N$$

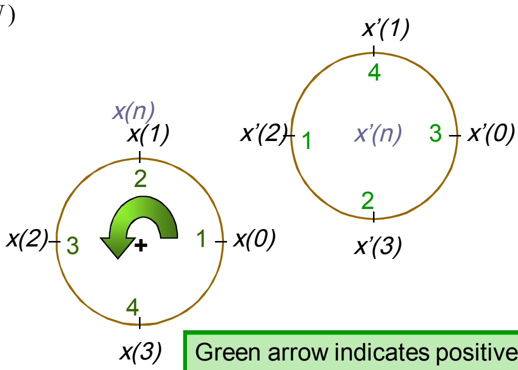
For our example

$$x'(0) = x((-2))_4 = x(2)$$

$$x'(1) = x((-1))_4 = x(3)$$

$$x'(2) = x((0))_4 = x(0)$$

$$x'(3) = x((1))_4 = x(1)$$



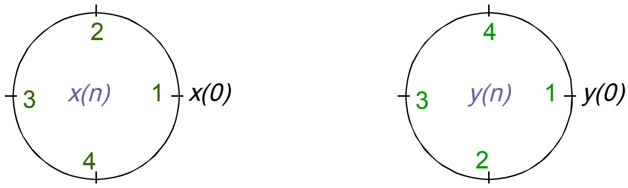
Green arrow indicates positive direction

Circular Reflection

The time reversal of an N-point sequence is attained by revereseing it samples about the point zero on the circle.

$$x((-n))_N = x(N-n)$$

Example $y(n) = x((-n))_N$

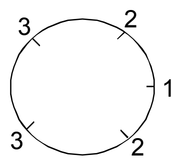


Circular Symmetry

An N-point sequence is called **circularly even** if

$$x(N-n) = x(n) \qquad 1 \le n \le N-1$$

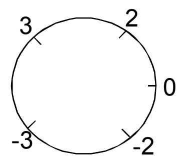
Example : $x(n)$ = {1,2,3,3,2}



An N-point sequence is called **circularly odd** if

$$x(N-n) = -x(n) \qquad 1 \le n \le N-1$$

Example : $x(n)$ = {0,2,3, -3, 2}



For circularly odd signal $x(0) = 0$ and if N is even $x(N/2) = 0$.

Circular Convolution

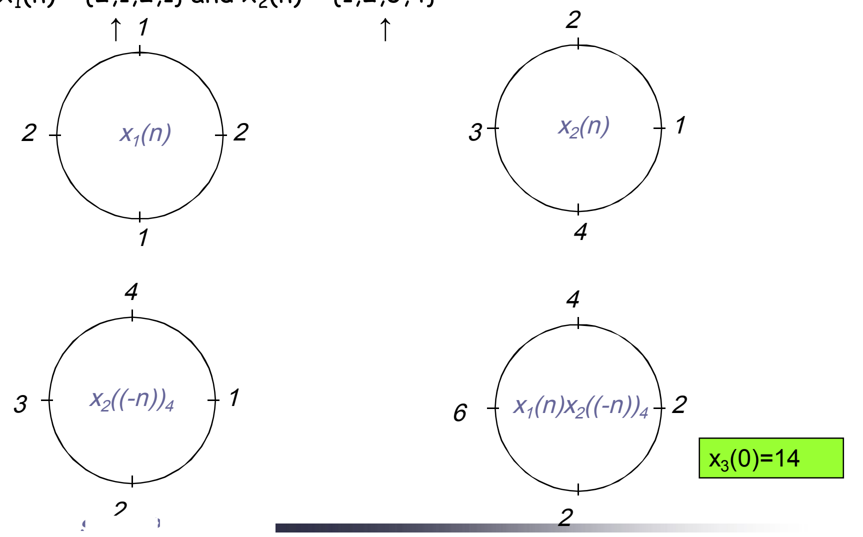
The circular convolution is very similar to normal convolution apart from that the signal is shifted using circular shift.

If $x_1(n)$ and $x_2(n)$ is two discrete signals, then the result of the circular convolution $x_3(n)$ can be realised by

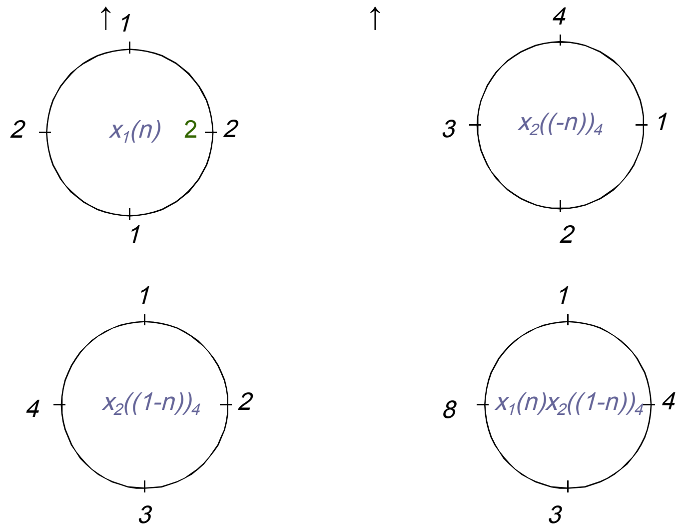
$$x_3(m) = x_1(n) \otimes x_2(n)$$
$$= \sum_{n=0}^{N-1} x_1(n) x_2((m-n))_N$$

Example: Perform the circular convolution of the following two sequences

$x_1(n) = \{2,1,2,1\}$ and $x_2(n) = \{1,2,3,4\}$

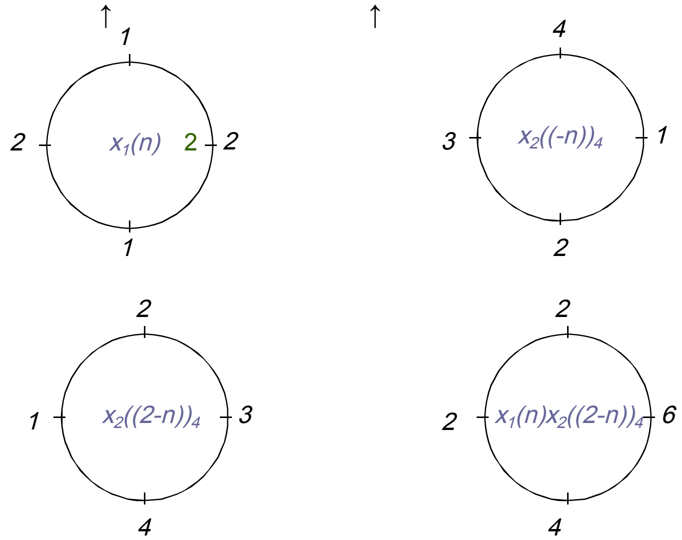


$x_1(n) = \{2,1,2,1\}$ and $x_2(n) = \{1,2,3,4\}$



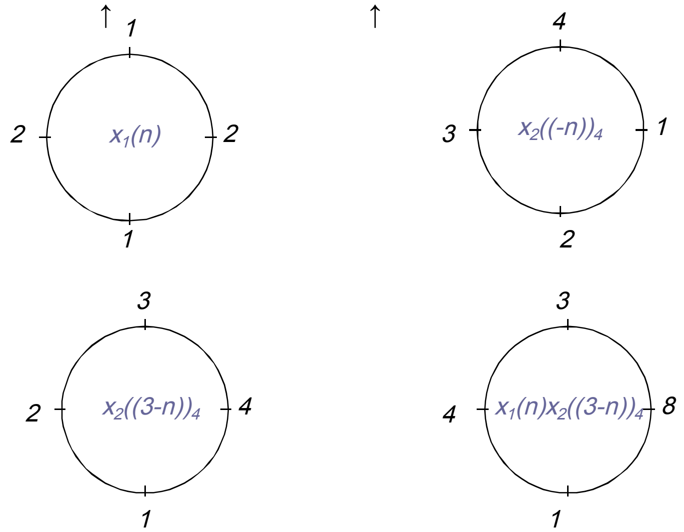
$x_3(1)=16$

$x_1(n) = \{2,1,2,1\}$ and $x_2(n) = \{1,2,3,4\}$



$x_3(2)=14$

$x_1(n) = \{\underset{\uparrow}{2},1,2,1\}$ and $x_2(n) = \{\underset{\uparrow}{1},2,3,4\}$



$x_3(3)=16$

$x_1(n) = \{\underset{\uparrow}{2},1,2,1\}$ and $x_2(n) = \{\underset{\uparrow}{1},2,3,4\}$

The final solution is

$$x_3(m) = x_1(n) \otimes x_2(n) = \{\underset{\uparrow}{14},16,14,16\}$$

References:

[1]. Dimitris G Manolakis, John G. Proakis, Digital Signal Processing: Principles, Algorithms, and Applications, 3rd edition.

[2]. Dimitris G. Manolakis, Vinay K. Ingle, Applied Digital Signal Processing: Theory and Practice.