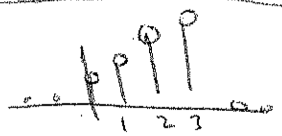
2016

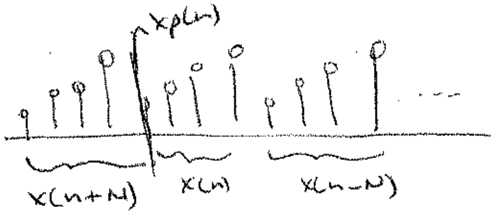
5

Relationship between $x(n)$ and $x_p(n)$

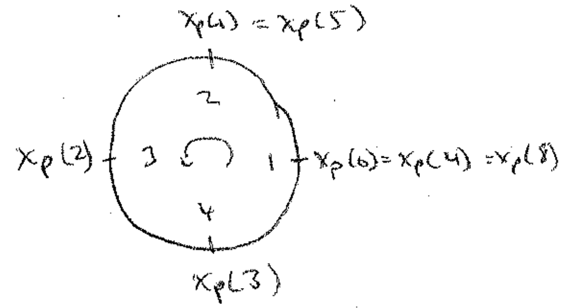
$x(n) = \{ \underset{\uparrow}{1}, 2, 3, 4 \}$ =  $L = 4$

now, say $x(n)$ is one period of a signal $x_p(n)$ that has period $N \geq L$. How do we picture this?

Case 1: $N = L$



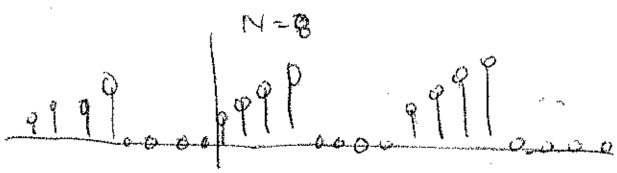
equivalent picture

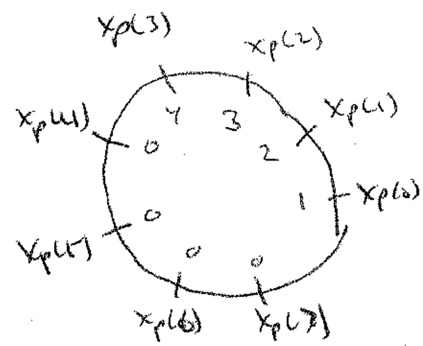


This is a handy way to think of the periodic signal. Time goes counter-clockwise.

Case 2: $N > L$

here, we "pad" $x(n)$ with zeros





why not always $N = L$?

→ sometime have 2 signals of different lengths

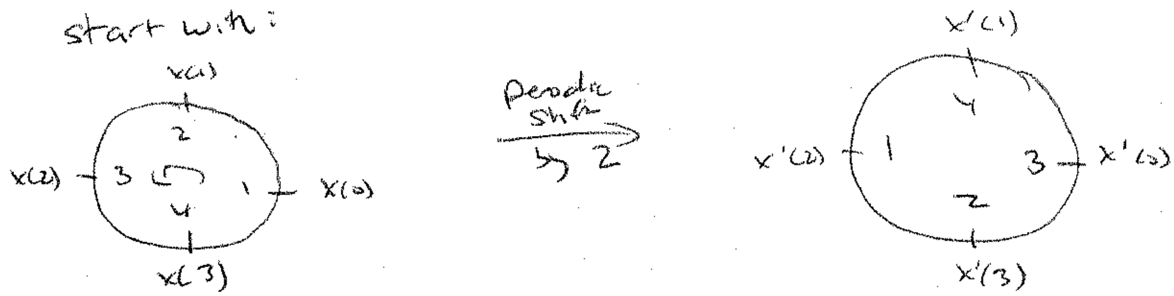
→ see example 7.1.2 in book / class slides

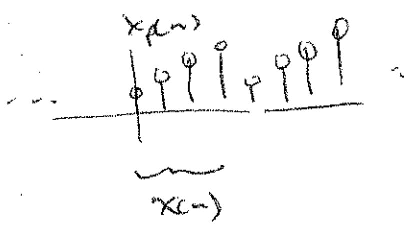
Shifting periodic sequences

→ For linear time shift (used before) samples just slide over.

→ for a periodic time shift, now signals rotate in:

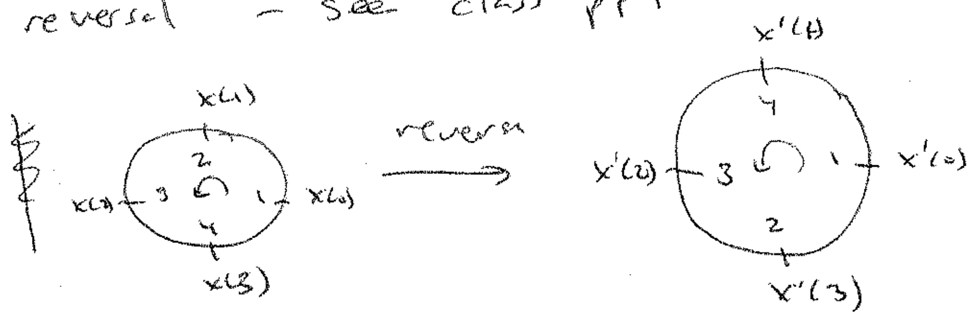
start with:





$x'(n) = \{3, 4, 1, 2\}$

Time reversal - see class ppt



Review from last lecture:

We saw that a periodic signal $x_p(n)$ could be thought of as a circular sequence:

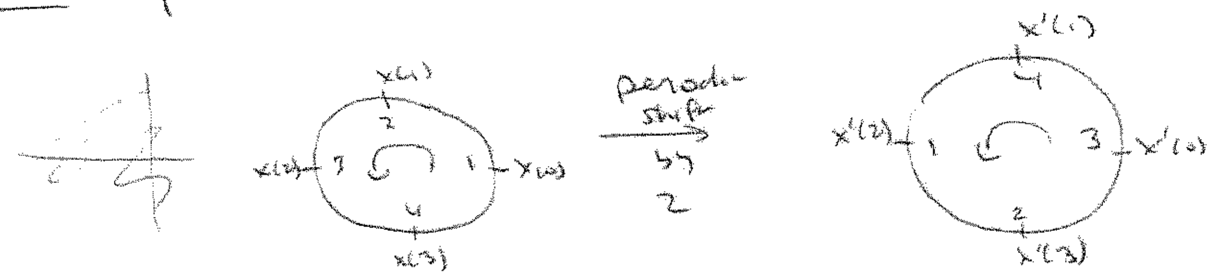
math to go with this:

$$x_p(n) = x(n \text{ modulo } N)$$
$$\equiv x((n))_N$$

examples:

0 mod 3 = 0
2 mod 3 = 2
3 mod 3 = 0
4 mod 3 = 1
etc.

Review: periodic shift



(in ppt)

+ time reversal (PPT)

check time-reversed $x' = x(N-n)$, $0 \le n \le N-1$

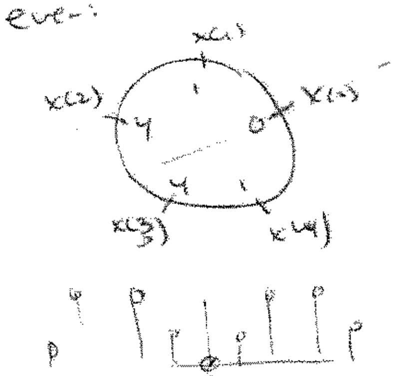
if $N=4$, $x'(0) = x(4) = x(0)$
$x'(1) = x(4-1) = x(3)$
$x'(2) = x(4-2) = x(2)$
etc.

Properties: even and odd

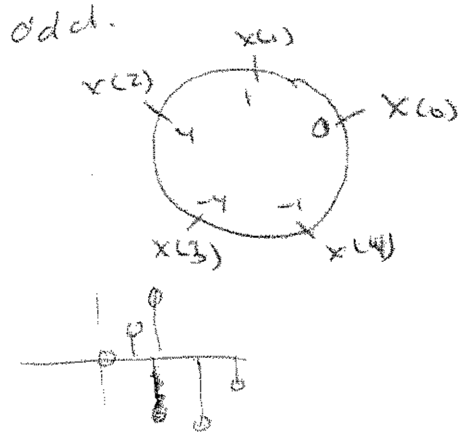
even signal: $x(N-n) = x(n)$ $\quad 0 \le n \le N-1$

odd signal $x(N-n) = -x(n)$ $\quad$ "

even:



odd.



DFT properties - P&M 7.2, table 7.2

Just as w/ CTFT & DTFT, we can define properties of DFT. Most are very familiar, but have some adjustment for circular periodic nature of DFT

Symmetry properties: For real $x(n)$

$$X(N-k) = X^*(k)$$ "circular Hermitian symmetry"

$$|X(N-k)| = |X(k)|$$ = circular even

$$\angle X(N-k) = -\angle X(k)$$ - circular odd.

where do these come from:

- real & even $x(n)$ $\rightarrow$ Real, even $X(k)$
- real & odd $x(n)$ $\rightarrow$ imag, odd $X(k)$

③ CCL

Properties – ① time reversal

If $x(n) \Longleftrightarrow X(k)$, then

$x((-n))_N = x(N-n) \Longleftrightarrow X((-k))_N = X(N-k)$

(like DTFT: $x(-n) \Leftrightarrow X(-\omega)$)

Proof :

~~$DFT\{x(N-m)\} = \sum_{n=0}^{N-1}$~~ put in PPT

② Circular time shift

if $x(n) \Leftrightarrow X(k)$, then

$x((n-\ell))_N \Longleftrightarrow X(k)\, e^{-j2\pi k\ell/N}$

proof: see PoM

③ similarly, circular freq shift

$x(n)\, e^{j2\pi \ell n/N} \Leftrightarrow X((k-\ell))_N$

④ Parseval's: $\sum_{k=0}^{N-1} |x(n)|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X(k)|^2$

⑤ the big one: convolution / multiplication duality

(in PPT)

Final result $x_1(n) \text{ Ⓝ } x_2(n) \Longleftrightarrow X_1(k) X_2(k)$

look at on-line applets for intuition – also

PoM 7.2.2

(4) CCL

Doing circular convolution (N)

time domain

a) locate $x_1(m)$ ~~x_2 ccw~~ on N-point circle

b) " ~~$x_2(m)$~~ $x_2((-m))_N$ " " " "

c) for $n = 0, 1, N-1$

1) rotate x_2 by one point CCW

2) multiply

3) add

freq domain (DFT)

1) find $X_1(k)$ for $k = 0, 1, .. N-1$

2) " $X_2(k)$ " " "

3) find $X_3(k) = X_1(k) X_2(k)$, $k = 0, ...$

4) $x_3(n) = F^{-1}\{X_3(k)\}$ for $n = 0, 1, .. N-1$

$x = [1\ 2]$

$y = [$ ~~4 5 6~~
3 4 5]

what is

x (3) y = 13 10 13

~~$x*x$~~

$x * y =$ 3 10 13 10

If $x = [1\ 2]$

$y = [3\ 4\ 5]$

~~what is x (3)~~

a) draw out circles

b) do computation

c) how long is linear conv $x * y$

7.3 use of DFT in linear filtering

we want to find $y(n) = h(n) * x(n)$
↑ regular convolution

we know
1) $Y(\omega) = H(\omega) X(\omega)$
so $y(n) = F^{-1}(H(\omega) X(\omega))$

we also know that if $x(n)$ is length L,
$h(n)$ is length M, then

$y = h * x$ is of length $L+M-1$ //

* thus if we want to find $Y(\omega)$, we need a number of points in the DFT $N \geq (L+M-1)$
- using fewer points will mean we've undersampled $Y(\omega)$. Then, $y(n) = F^{-1}(Y(\omega))$ will have <u>time-domain aliasing</u>.
- using enough samples means $h \circledN x = h * x$

If we handle this ok, we can use DFT to do linear convolution as follows:

1) given $x(n)$ of length L
$h(n)$ of length M
2) set $N = L+M-1$
3) zero-pad $x(n)$ so it's of length N $(M-1 \text{ zeros})$
" " h " " " " " $(L-1 \text{ zeros})$
4) take N-point DFT's of x, h
5) $y(n) = DFT^{-1}\{Y(k)\}$

6 CCL

- go through on-line examples
- go thru example on board

DFT Wrapup / FFT Lecture

Topic 1: review derivation of $X_1(k) X_2(k) \longleftrightarrow x_1 \, (N) \, x_2$
(in PPT)

Review last time

→ Algorithm for circular convolution: place x_1 and $x_2((-n))_N$ on circle, rotate x_2 counter-clockwise, multiply, add

→ avoiding time domain aliasing
- if $x(n)$ is length L, $h(n)$ is length M, then linear convolution output is length $L+M-1 = N$
- thus, the DFT of $y(n) = x(n) * h(n)$ must have
  $N \geq L+M-1$
- thus, we must zero-pad x and h to avoid time-domain aliasing; then,
  $x \, (N) \, h = x * h$ for 1st N points
- ~~finally~~ $y(n) = DFT^{-1}(X(k)H(k))$ in frequency domain

DFT-based filtering of long sequences

If $x(n)$ is a long sequence, h is short, then:
1) break $x(n)$ into short blocks, length L
2) use DFT to filter each block
3) paste things back together

Overlap-add strategy (figure is in PPT)

algorithm: for block $\alpha = 1, 2, 3, \ldots$

1) take a length L block of data
2) zero-pad with M-1 ~~zeros~~ zeros ~~[illegible]~~
   so for first block,
   $x_\alpha = [x(1), \ldots x(L), 0, 0 \ldots 0]$ ← M-1 zeros

3) zero pad h(n) so it's length N
   $h = [h(1), h(2) \ldots h(M), 0, 0, 0, 0 \ldots 0]$ ← L-1

4) find $\tilde{Y}_\alpha(k) = X_\alpha(k) H(k)$ ← no subscript; h always the same.
   and $\tilde{y}_\alpha = \text{IDFT}^{-1}\{X_\alpha(k) H(k)\}$

5) problem: the last M-1 samples of $\tilde{y}_\alpha(n)$ do not include the effects of the first M-1 samples in next block.
   solution: add last M-1 points of $\tilde{y}_\alpha(n)$ to first M-1 points of $\tilde{y}_{\alpha+1}(n)$

Overlap-save strategy (figure is in PPT)

algorithm for block $\alpha = 1, 2, 3 \ldots$

1) take length L data block
2) put previous (M-1) points at beginning
   (for $\alpha = 1$, pre-pad w/ M-1 zeros)
3) thus input length $N = L+M-1$; ~~for length~~ do an N-point DFT
   - note we'd really want a $L + (M-1)\times 2 = L+2M-2$ length filter; since we are short, first M-1 points are time aliased

thus N - point DFT gives us

$$\tilde{Y}_\alpha(k) = \underline{V}_\alpha(k)\,H(k) \quad , N = M+L-1 \text{ points}$$

$$\tilde{y}_\alpha(n) = \underset{M+L-1}{\underset{\uparrow}{\mathrm{IDFT}}}\{\underline{V}_\alpha(k)\,H(k)\}$$

4) because we used a DFT that was too short, the first M-1 points of $\tilde{y}_\alpha(n)$ are aliased.

(5) Thus we discard the first M-1 points, keep rest, and continue.

Both overlap-add and overlap-sum process L data points in each block using an N = L+M-1 DFT. Thus, choice is really personal preference.