BINGHAMTON UNIVERSITY
State University of New York

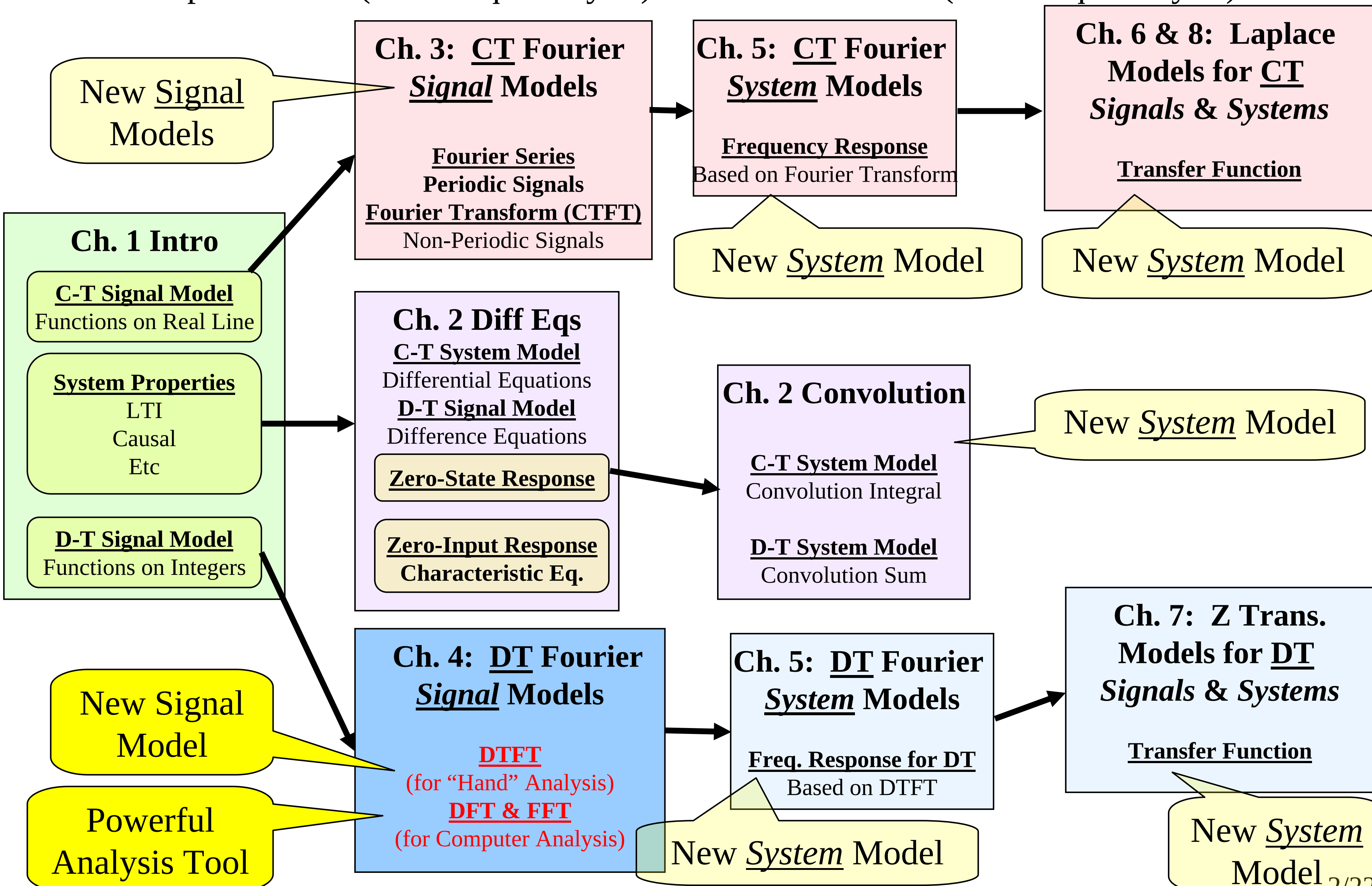
EECE 301
Signals & Systems
Prof. Mark Fowler

Note Set #25

- D-T Signals: Relation between DFT, DTFT, & CTFT
- Reading Assignment: Sections 4.2.4 & 4.3 of Kamen and Heck

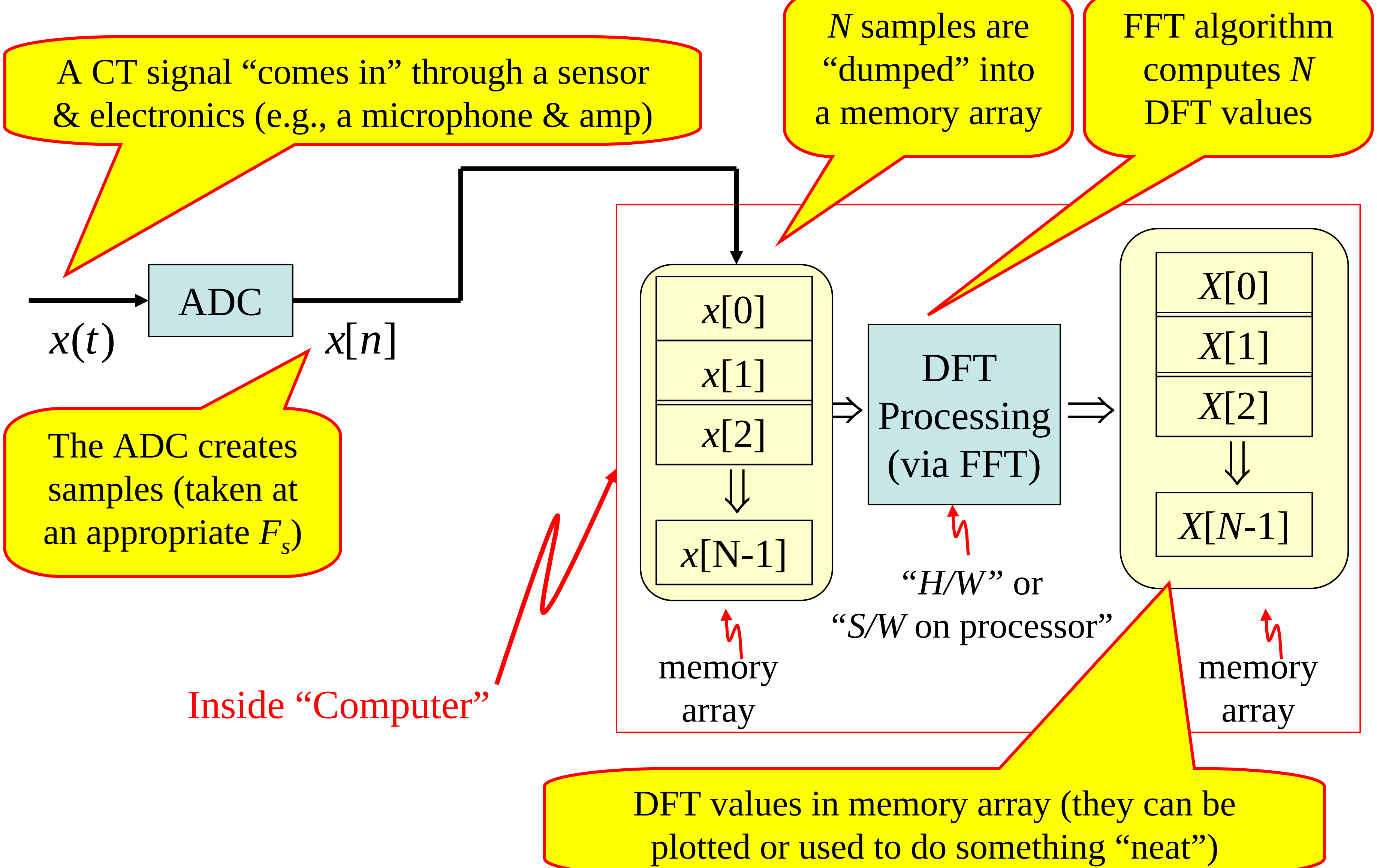
Course Flow Diagram

The arrows here show conceptual flow between ideas. Note the parallel structure between the pink blocks (C-T Freq. Analysis) and the blue blocks (D-T Freq. Analysis).



<u>We can use the DFT to implement numerical FT processing</u>

This enables us to <u>numerically</u> analyze a signal to find out what frequencies it contains!!!



If we are doing this DFT processing to see what the original CT signal $x(t)$ “looks” like in the frequency domain…

… we want the DFT values to be “representative” of the CTFT of $x(t)$

Likewise…

If we are doing this DFT processing to do some “neat” processing to extract some information from $x(t)$ or to modify it in some way…

… we want the DFT values to be “representative” of the CTFT of $x(t)$

So… we need to understand what the DFT values tell us about the CTFT of $x(t)$…

We need to understand the relations between…

CTFT, DTFT, and DFT

We'll mathematically explore the link between DTFT & DFT in two cases:

1. For $x[n]$ of **finite duration**:

$$\ldots 0 \;\; 0 \;\; \underbrace{x[0] \;\; x[1] \;\; x[2] \ldots \;\; X[N-1]}_{N \text{ “non-zero” terms}} \;\; 0 \;\; 0$$

(of course, we could have some of the interior values = 0)

This case hardly ever happens… but it's easy to analyze and provides a perspective for the 2nd case

For this case… we'll assume that the signal ***is*** zero outside the range that we have captured.

So… we have all of the meaningful signal data.

This is the practical case.

2. For $x[n]$ of **infinite duration** …or at least of duration longer than what we can get into our "DFT Processor" inside our "computer".

So… we don't have all the meaningful signal data.

What effect does that have? How much data do we need for a given goal?

DFT & DTFT: Finite Duration Case

If $x[n] = 0$ for $n < 0$ and $n \geq N$ then the DTFT is:

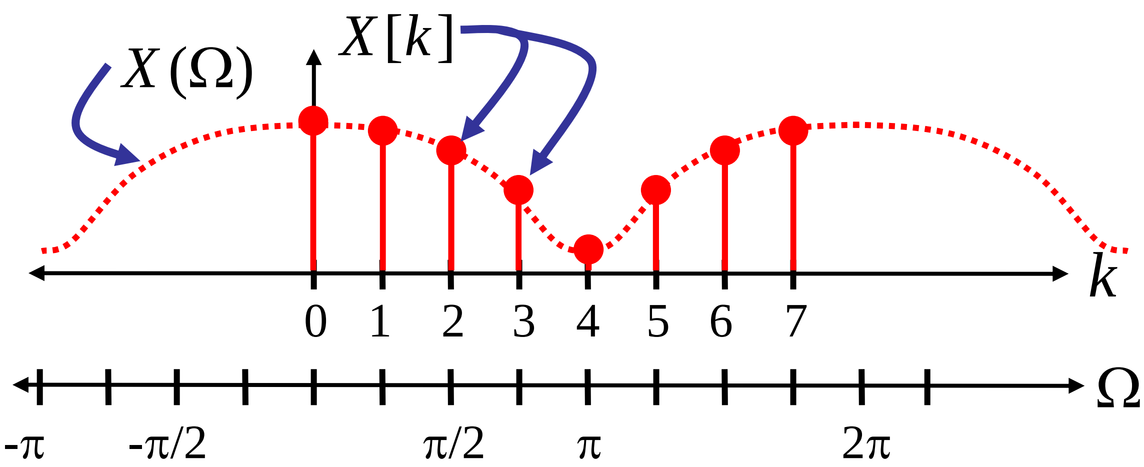
$$\boxed{X(\Omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n} = \sum_{n=0}^{N-1} x[n]e^{-j\Omega n}}$$

we can leave out terms that are zero

Now… if we take these N samples and compute the DFT (using the FFT, perhaps) we get:

$$\boxed{X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N} \qquad k = 0,1,2,\ldots,N-1}$$

Comparing these we see that for the finite-duration signal case: $\boxed{X[k] = X\left(k\frac{2\pi}{N}\right)}$



DTFT & DFT :

DFT points lie exactly on the finite-duration signal's DTFT!!!

Summary of DFT & DTFT for a ***finite*** duration $x[n]$

$$x[n] \xrightarrow{\text{DTFT}} X(\Omega)$$

$$x[n] \xrightarrow{\text{DFT}} X[k] = X\left(k\frac{2\pi}{N}\right)$$

Points of DFT are "samples" of DTFT of *x*[*n*]

The number of samples N sets how closely spaced these "samples" are on the DTFT… seems to be a limitation.

"Zero-Padding Trick"

After we collect our N samples, we tack on some additional zeros at the end to trick the "DFT Processing" into thinking there are really more samples.

(Since these are zeros tacked on they don't change the values in the DFT sums)

If we now have a total of N_Z "samples" (including the tacked on zeros), then the spacing between DFT points is $2\pi/N_Z$ which is smaller than $2\pi/N$

Ex. 4.11 DTFT & DFT of pulse

$$x[n] = \begin{cases} 1, & n = 0,1,2,\ldots 2q \\ 0, & otherwise \end{cases}$$

Recall: $p_q[n] = \begin{cases} 1, & n = -q, \ldots, -1, 0, 1, \ldots, q \\ 0, & otherwise \end{cases}$

Note: we'll need the delay property for DTFT

Then… $x[n] = p_q[n-q]$

From DTFT Table: $\boxed{p_q[n] \leftrightarrow P_q(\Omega) = \dfrac{\sin[(q+0.5)\Omega]}{\sin[\Omega/2]}}$

From DTFT Property Table

(Delay Property): $\boxed{X(\Omega) = \dfrac{\sin[(q+0.5)\Omega]}{\sin[\Omega/2]} e^{-jq\Omega}}$

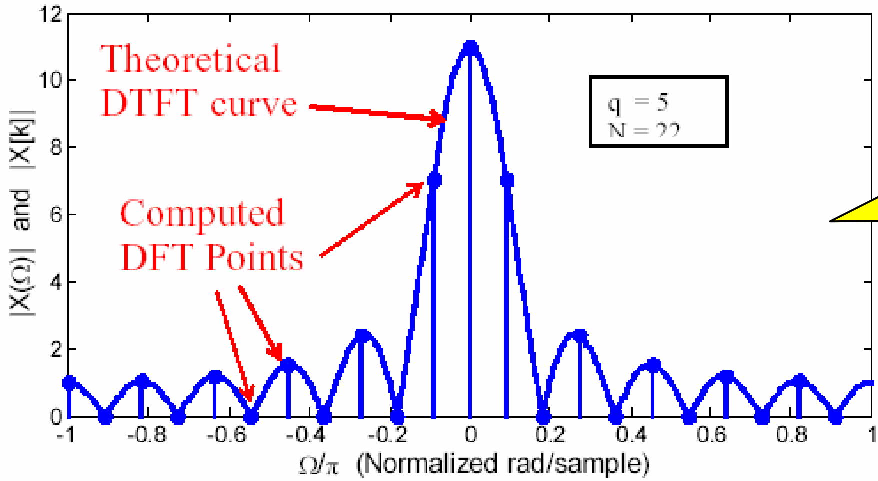
Since $x[n]$ is a finite-duration signal then the DFT of the $N = 2q+1$ non-zero samples is just samples of the DTFT: $\boxed{X[k] = X\left(k\dfrac{2\pi}{N}\right)}$

$$\boxed{X[k] = \frac{\sin[(q+.5)2\pi k/N]}{\sin[\pi k/N]} e^{-jq2\pi k/N}}$$

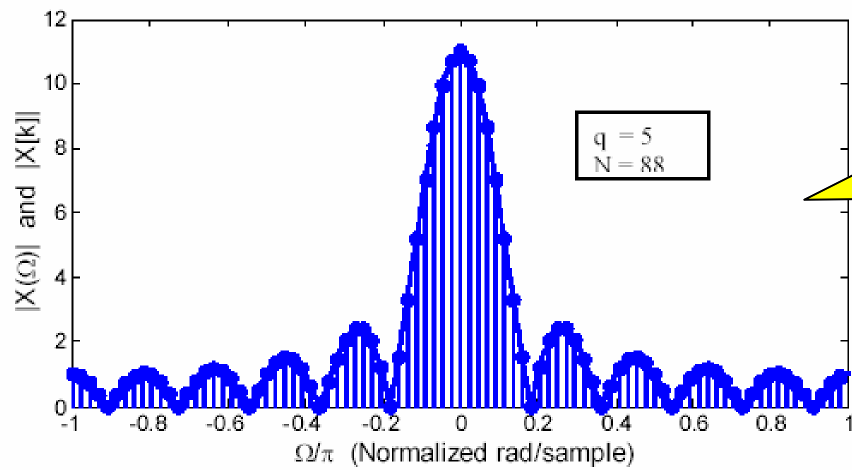
Note that if we don't zero pad, then all but the $k = 0$ DFT values are zero!!!

That doesn't show what the DTFT looks like! So we need to use zero-padding.

Here are two numerically computed examples, both for the case of $q = 5$:



For the case of zero-padding 11 zeros onto the end of the signal… the DFT points still don't really show what the DTFT looks like!



For the case of zero-padding 77 zeros onto the end of the signal… **NOW** the DFT points really show what the DTFT looks like!

DFTs were computed using matlab's fft command… see code on next slide

```
omega=eps+(-1:0.0001:1)*pi;
q=5;    % used to set pulse length to 11 points
X=sin((q+0.5)*omega)./sin(omega/2);
subplot(2,1,1)
plot(omega/pi,abs(X));   % plot magn of DTFT
xlabel('\Omega/\pi  (Normalized rad/sample)')
ylabel('|X(\Omega)|  and  |X[k]|')
hold on
x=zeros(1,22);  % Initially fill x with 22 zeros
x(1:(2*q+1))=1;   % Then fill first 11 pts with ones
Xk=fftshift(fft(x));    % fft computes the DFT and fftshift re-orders points
                        % to between  -pi and pi
omega_k=(-11:10)*2*pi/22;  % compute DFT frequencies, except make them
                        % between  -pi and pi
stem(omega_k/pi,abs(Xk));  % plot DFT vs. normalized frequencies
hold off
subplot(2,1,2)
plot(omega/pi,abs(X));
xlabel('\Omega/\pi  (Normalized rad/sample)')
ylabel('|X(\Omega)|  and  |X[k]|')
hold on
x=zeros(1,88);
x(1:(2*q+1))=1;
Xk=fftshift(fft(x));
omega_k=(-44:43)*2*pi/88;
stem(omega_k/pi,abs(Xk));
hold off
```

Compute the DTFT Equation derived for the pulse. Using *eps* adds a very small number to avoid getting $\Omega = 0$ and then dividing by 0

Make the zero-padded signal

Compute the DFT

Compute the DFT point's frequency values and plot the DFT

Important Points for *Finite-Duration* Signal Case

- DFT points lie on the DTFT curve… perfect view of the DTFT
  - But… only if the DFT points are spaced closely enough
- Zero-Padding doesn’t change the shape of the DFT…
- It just gives a denser set of DFT points… all of which lie on the true DTFT
  - Zero-padding provides a better view of this “perfect” view of the DTFT

DFT & DTFT: Infinite Duration Case

As we said… in a computer we cannot deal with an infinite number of signal samples.

So say there is some signal that "goes on forever" (or at least continues on for longer than we can or are willing to grab samples)

$x[n] \quad n = \ldots, -3, -2, -1, 0, 1, 2, 3, \ldots$

We **only grab N samples**: $x[n]$, $n = 0, \ldots, N-1$ **We've lost some information!**

We can define an "imagined" finite-duration signal:

$$x_N[n] = \begin{cases} x[n], & n = 0, 1, 2, \ldots, N-1 \\ 0, & elsewhere \end{cases}$$

We can compute the DFT of the N collected samples:

$$X_N[k] = \sum_{n=0}^{N-1} x_N[n] e^{-j2\pi nk/N} \qquad k = 0, 1, \ldots, N-1$$

Q: How does this DFT of the "truncated signal" relate to the "true" DTFT of the full-duration $x[n]$? …which is what we really want to see!!

$$\text{"True" DTFT}: X_\infty(\Omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n}$$

What we *want* to see

$$\text{DTFT of truncated signal}: X_N(\Omega) = \sum_{n=-\infty}^{\infty} x_N[n]e^{-j\Omega n} = \sum_{n=0}^{N-1} x[n]e^{-j\Omega n}$$

A *distorted* version of what we want to see

$$\text{DFT of collected signal data}: X_N[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}$$

What we *can* see

$$\boxed{\text{DFT gives samples of } X_N(\Omega)}$$

So… DFT of collected data gives "samples" of $\underbrace{\text{DFT of } \textit{truncated} \text{ signal}}_{\neq \text{ "True" DTFT}}$

⇒DFT of collected data does not perfectly show DTFT of complete signal.

Instead, the DFT of the data shows the DTFT of the *truncated* signal…

So **our goal** is to understand what kinds of "errors" are in the "truncated" DTFT
…then we'll know what "errors" are in the computed DFT of the data

To see what the DFT <u>does</u> show we need to understand how

$X_N(\Omega)$ relates to $X_\infty(\Omega)$

First, we note that:

$$x_N[n] = x[n]\underbrace{p_q[n-q]}$$

DTFT

$$P_q(\Omega) = \frac{\sin[N\Omega/2]}{\sin[\Omega/2]} e^{-j(N-1)\Omega/2}$$

with $N=2q+1$

From "mult. in time domain" property in DTFT Property Table:

$$\boxed{X_N(\Omega) = X_\infty(\Omega) * P_q(\Omega)}$$

Convolution causes "smearing" of $X_\infty(\Omega)$

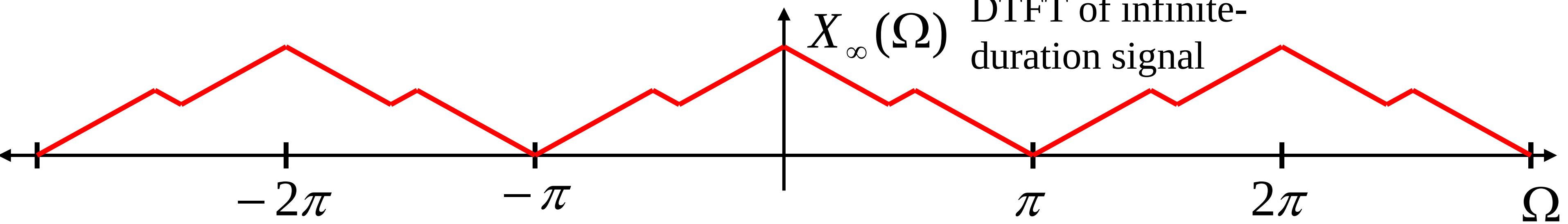
$\Rightarrow$ So… $X_N(\Omega)$ …which we can see via the DFT $X_N[k]$ …

is a "smeared" version of $X_\infty(\Omega)$

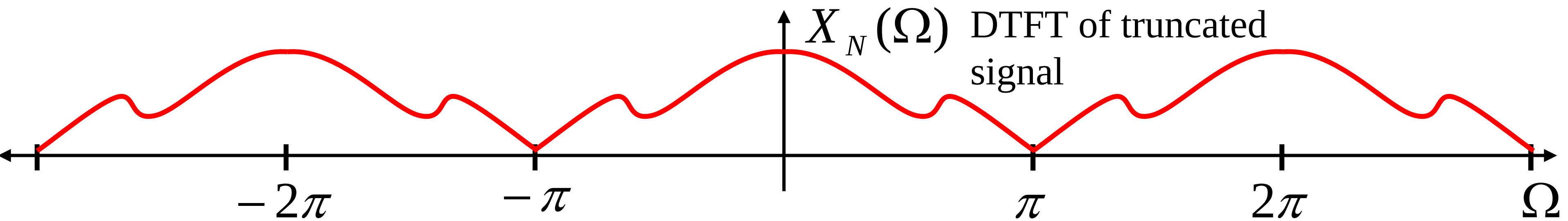
"Fact": The more data you collect, the less smearing

… because $P_q(\Omega)$ becomes more like $\delta(\Omega)$

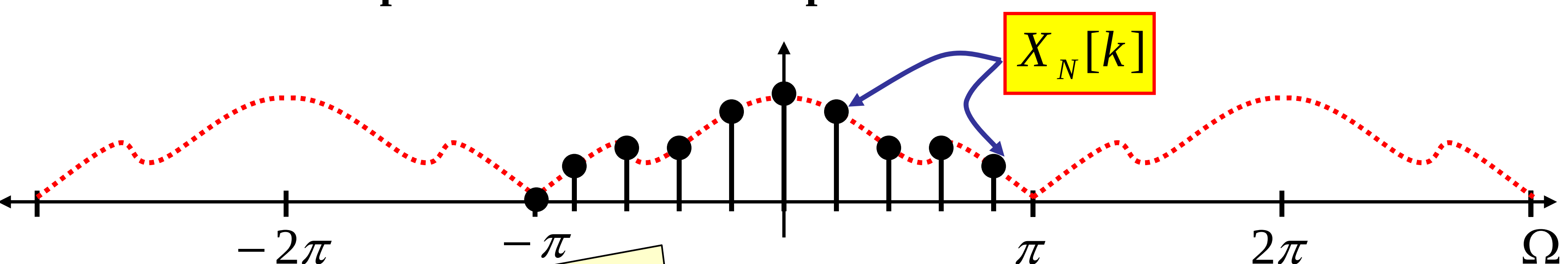
Suppose the infinite-duration signal's DTFT is:



Then it gets smeared into something that might look like this:



Then the DFT computed from the *N* data points is:



The DFT points are shown *after* "upper" points are moved (e.g., by matlab's "fftshift"

Example: Infinite-Duration Complex Sinusoid & DFT

Suppose we have the signal $x[n] = e^{-j\Omega_o n}$ $n = \ldots, -3, -2, -1, 0, 1, 2, \ldots$

and we want to compute the DFT of N collected samples (n = 0, 1, 2, …, N-1).

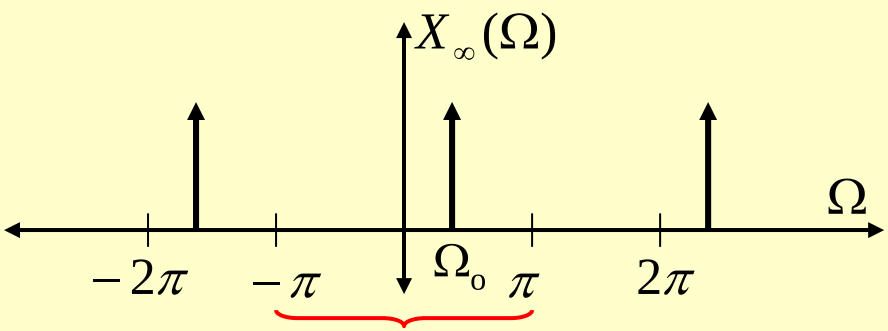
This is an important example because in practice we often have signals that consists of a few significant sinusoids among some other signals (e.g. radar and sonar).

In practice we just get the N samples and we compute the DFT… but before we do that we need to understand what the DFT of the N samples will show.

So we first need to theoretically find the DTFT of the infinite-duration signal.

From DTFT Table we have:

$$X_\infty(\Omega) = \begin{cases} \delta(\Omega - \Omega_0), & -\pi < \Omega < \pi \\ periodic\ \ elsewhere \end{cases}$$



From our previous results we know that the DTFT of the collected data is:

$$X_N(\Omega) = X_\infty(\Omega) * \left[\frac{\sin[N\Omega/2]}{\sin[\Omega/2]} e^{-j(N-1)\Omega/2} \right]$$

$P_q(\Omega)$

Just Delta's in here $\Rightarrow$ Use Sifting Property!!

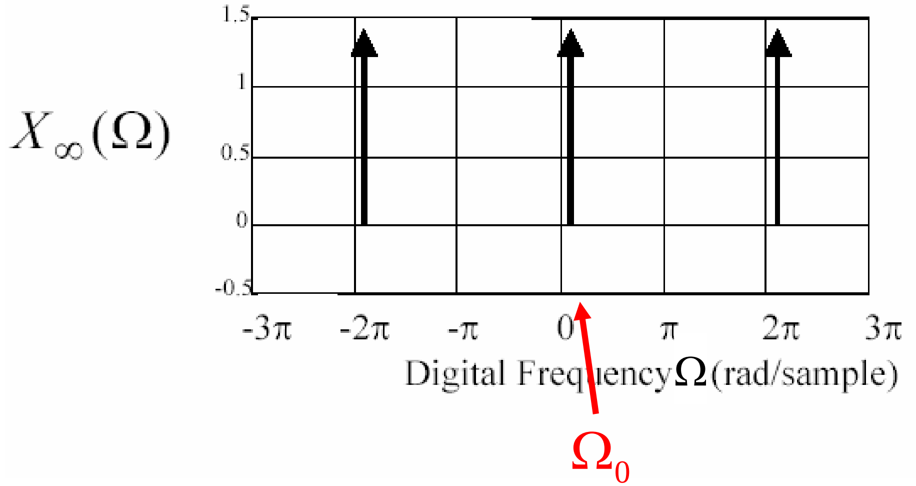
$$X_\infty(\Omega) = \begin{cases} \delta(\Omega - \Omega_0), & -\pi < \Omega < \pi \\ \textit{periodic elsewhere} \end{cases}$$

Just a shifted version of $P_q(\Omega)$

$$X_N(\Omega) = \begin{cases} \dfrac{\sin\left[\dfrac{N(\Omega-\Omega_0)}{2}\right]}{\sin\left[\dfrac{(\Omega-\Omega_0)}{2}\right]} e^{-j(N-1)(\Omega-\Omega_0)/2}, & -\pi < \Omega < \pi \\ \textit{periodic elsewhere} \end{cases}$$

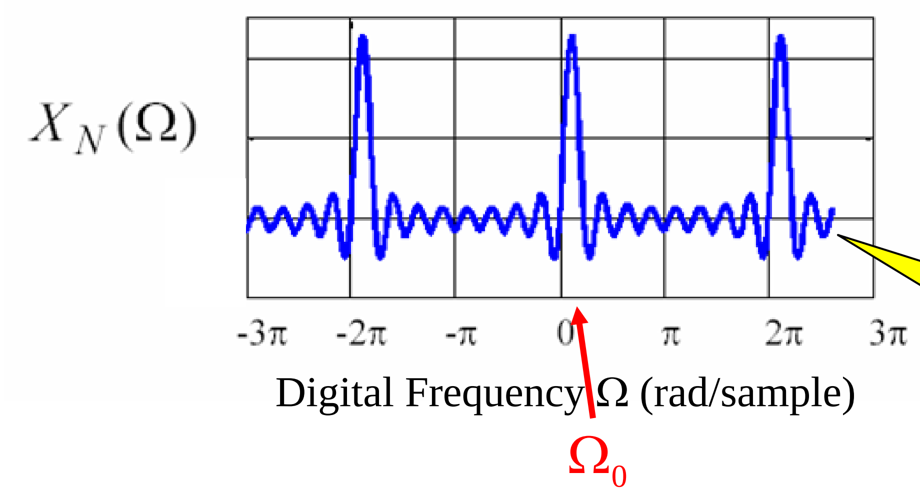
This is the DTFT on which our data-computed DFT points will lie… so looking at this DTFT shows us what we can expect from our DFT processing!!!

True DTFT of Infinite Duration Complex Sinusoid



$$X_\infty(\Omega) = \begin{cases} \delta(\Omega - \Omega_0), & -\pi < \Omega < \pi \\ periodic \;\; elsewhere \end{cases}$$

DTFT of Finite Number of Samples of a Complex Sinusoid

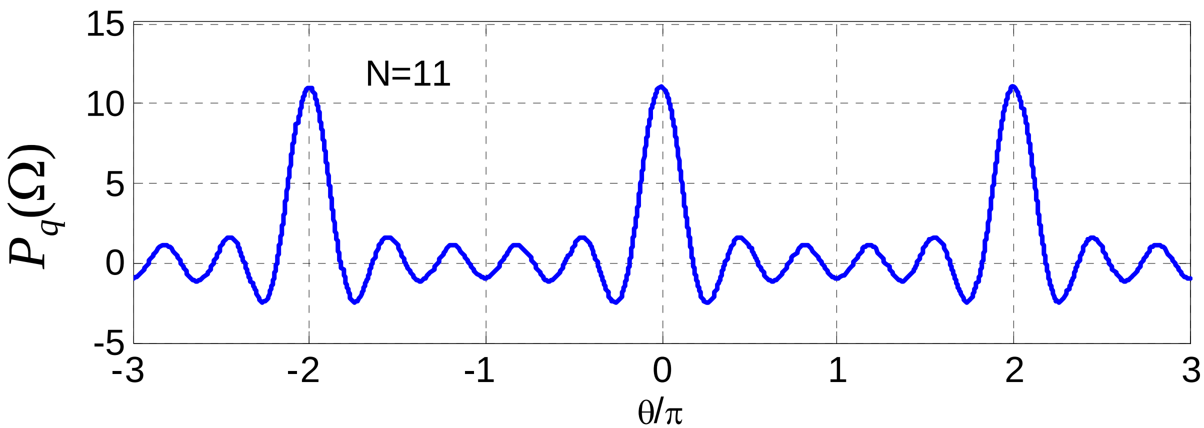


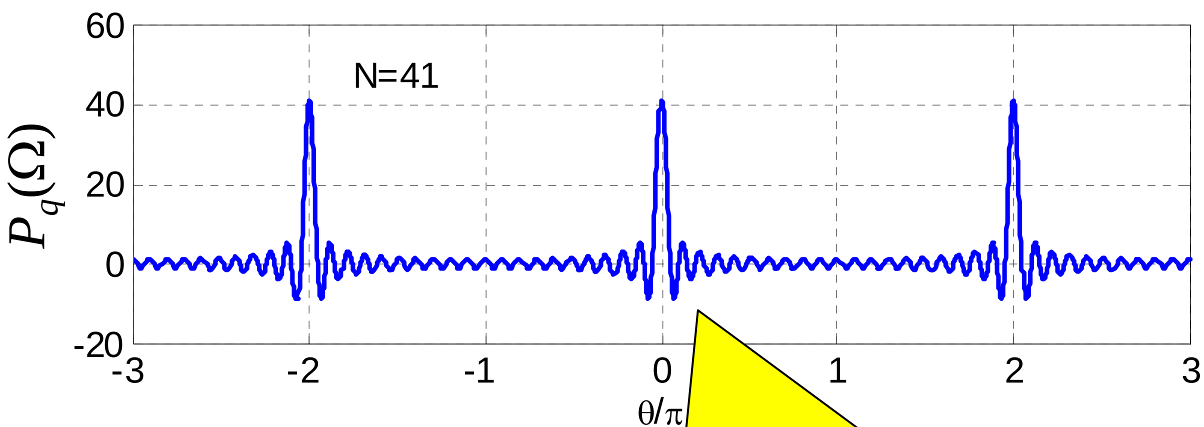
$$X_N(\Omega) = \begin{cases} \dfrac{\sin\left[\dfrac{N(\Omega-\Omega_0)}{2}\right]}{\sin\left[\dfrac{(\Omega-\Omega_0)}{2}\right]} e^{-j(N-1)(\Omega-\Omega_0)/2}, & -\pi < \Omega < \pi \\ periodic \; elsewhere \end{cases}$$

The computed DFT would give points on this curve… the spacing of points is controlled through "zero padding"

So… what effect does our choice of N have???

To answer that we can simply look at $P_q(\Omega)$ for different values of $N = 2q+1$ …



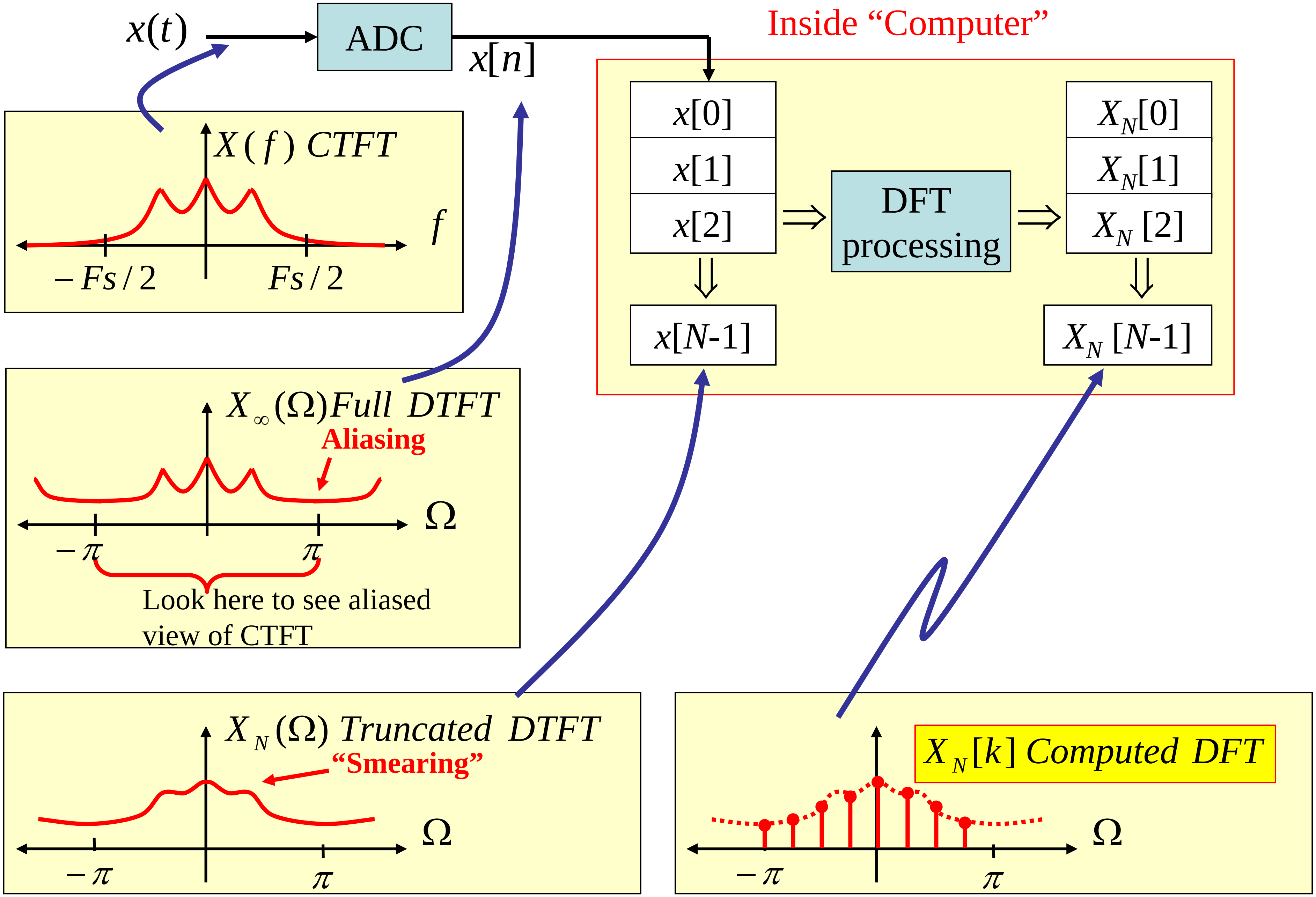


As N grows… looks more like a delta!!
So… less smearing of $X_N(\Omega)$!!

Important points for Infinite-Duration Signal Case

1. DTFT of finite collected data is a "smeared" version of the DTFT of the infinite-duration data
2. The computed DFT points lie on the "smeared" DTFT curve… not the "true" DTFT
   a. This gives an imperfect view of the true DTFT!
3. "Zero-padding" gives denser set of DFT points… a better view of this imperfect view of the desired DTFT!!!

Connections between the CTFT, DTFT, & DFT



Errors in a Computed DFT

CTFT

↓

$DTFT_\infty$

↓

$DTFT_N$

↓

DFT

Aliasing Error – control through F_s choice
(i.e. through proper sampling)

See DSP course

"Smearing" Error – control through $\begin{cases} N \text{ choice} \\ \text{"window" choice} \end{cases}$

"Grid" Error – control through $\begin{cases} N \text{ choice} \\ \text{"zero padding"} \end{cases}$

This is the only thing we can compute from data… and it has all these "errors" in it!! The theory covered here allows an engineer to understand how to control the amount of those errors!!!

Zero padding trick

Collect N samples $\rightarrow$ defines $X_N(\Omega)$

Tack M zeros on at the end of the samples

Take $(N + M)$pt. DFT $\rightarrow$ gives points on $X_N(\Omega)$ spaced by $2\pi/(N+M)$
(rather than $2\pi/N$)