BINGHAMTON
UNIVERSITY
State University of New York

EEO 401
Digital Signal Processing
Prof. Mark Fowler

Note Set #22

- Using the DFT for Spectral Analysis of Signals
- Reading Assignment: Sect. 7.4 of Proakis & Manolakis
  Ch. 6 of Porat's Book

Goal of Practical Spectral Analysis

Goal: Given a discrete-time signal $x[n]$, use DFT (via FFT) to analyze its spectral content – in particular, to detect the presence of sinusoids and estimate their frequency.

Challenges:

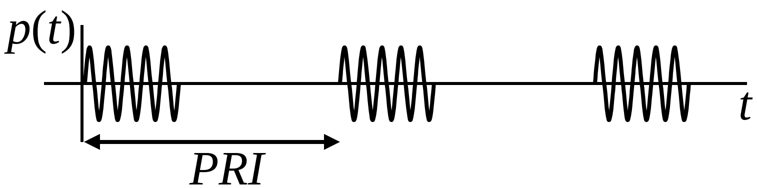
1. Available signal may be short (e.g., a radar signal may only be "on" for a very short time).
2. If the signal is long, then the spectral content may change with time (e.g., music spectrum changes with time) – so spectrum may be considered to be constant only a block-by-block basis where the blocks are short.

Both of these drive the need to apply the DFT to a short signal record ➔ **Challenge = Resolution & Accuracy**

Example Application (Electronic Warfare)

Intercept T seconds of a Radar Pulse Train $p(t)$, Compute DFT, detect & estimate peaks to identify type of radar.

"Underlying" Pulse Train is Periodic ➔ Fourier Series

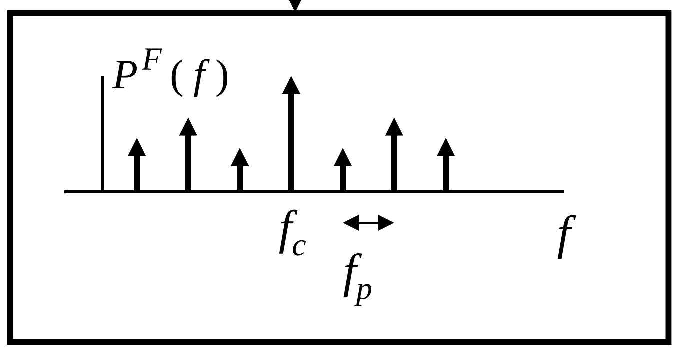


FS

$$p(t) = \sum_n c_n e^{jn2\pi(f_c - f_p)t}$$

$f_p = 1/PRI$

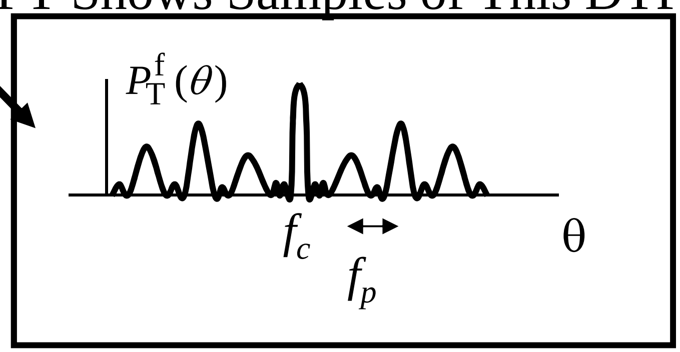
FT



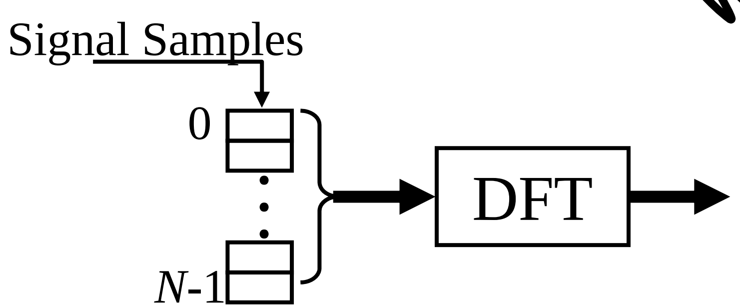
DTFT of **Infinite** Duration Signal

Since DFT shows samples of the DTFT of the finite duration signal we can study what the DFT gives us by looking at what the DTFT of a **finite-duration** signal looks like!!

DFT Shows Samples of This DTFT



DTFT of **Finite** Duration Signal



Effect of Windowing

Porat Sections 6.1 and 6.2

Basic Viewpoint of Signal Data

We are given a finite # of signal samples, and want to use them to see the spectrum of the infinite-duration signal….
How well can we do that?

Math Model for having a finite # of samples:

$$x[n] = \begin{cases} y[n], & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

Finite Duration Signal Available for Processing

Infinite Duration Signal

Better Math Model – Rectangular Window-Based Model:

$$x[n] = y[n]w_r[n], \quad \text{where } w_r[n] = \begin{cases} 1, & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

Implication of Window-Based Model

Since the available data $x[n]$ is related to the unavailable signal $y[n]$ through multiplication we can use the Multiplication Theorem for DTFT (Eq. (2.103) in Porat) to find out what we get!

Thus, the DTFT of the signal data is related to the DTFT of the infinite-duration signal by:

$$X^{\mathrm{f}}(\theta) = \frac{1}{2\pi}\left\{Y^{\mathrm{f}} \circledast W_r^{\mathrm{f}}\right\}(\theta) = \frac{1}{2\pi}\int_{-\pi}^{\pi} Y^{\mathrm{f}}(\lambda) W_r^{\mathrm{f}}(\lambda - \theta)\, d\lambda$$

where the DTFT of the rect. window is:

$$W_r^{\mathrm{f}}(\theta) = \sum_{n=0}^{N-1} 1 e^{-j\theta n} = \frac{1 - e^{-j\theta N}}{1 - e^{-j\theta}}$$

Use Geometric Sum

$$= \frac{e^{-j\theta N/2}\left[e^{j\theta N/2} - e^{-j\theta N/2}\right]/2j}{e^{-j\theta/2}\left[e^{j\theta/2} - e^{-j\theta/2}\right]/2j}$$

Use Euler!

$$= e^{-j\theta\frac{(N-1)}{2}} \frac{\sin(\theta N/2)}{\sin(\theta/2)}$$

$$D(\theta, N) \triangleq \frac{\sin(\theta N/2)}{\sin(\theta/2)}$$

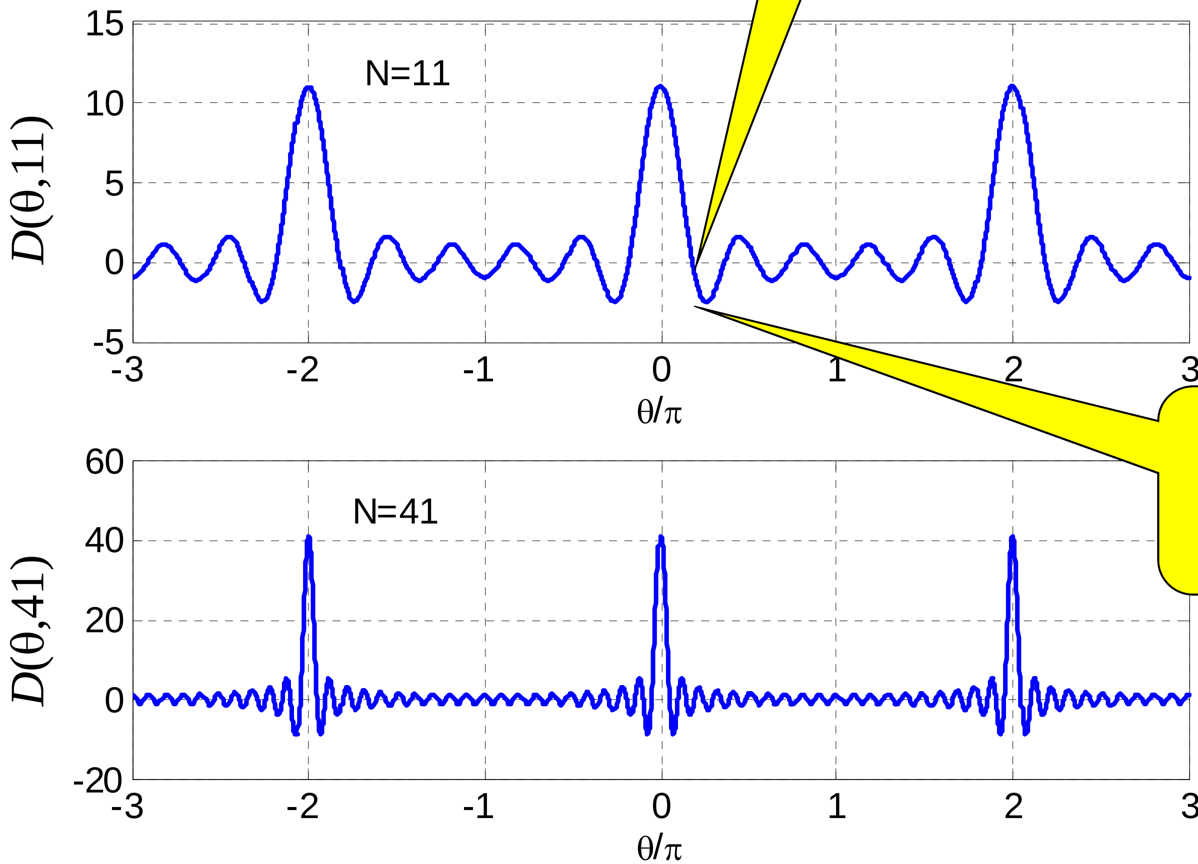
The “Dirichlet Kernel” $D(\theta,N)$

- Looks like “sinc”, except periodic
- Mainlobe Gets Narrower as $N\uparrow$
- Sidelobes “Get Lower” as $N\uparrow$
- Height of Mainlobe = N

➔ **Looks more like delta as $N\uparrow$**

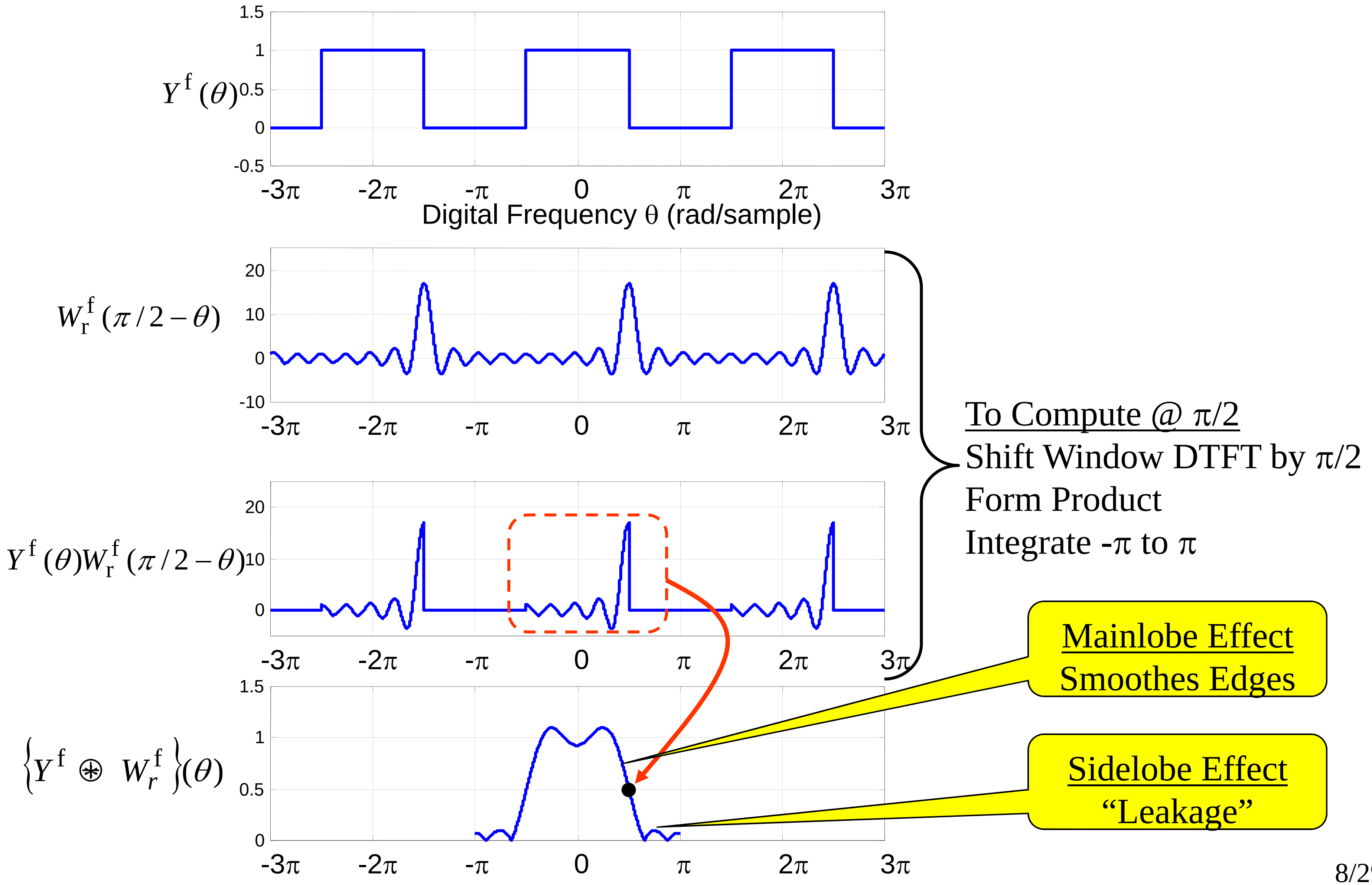
Nearest Zero
@ $\theta=2\pi/N$

→ Mainlobe Width = $4\pi/N$



Largest Sidelobe
–13 dB w.r.t. ML peak
(For Any N)

Impact of Window



Impact of Window (pt. 2)

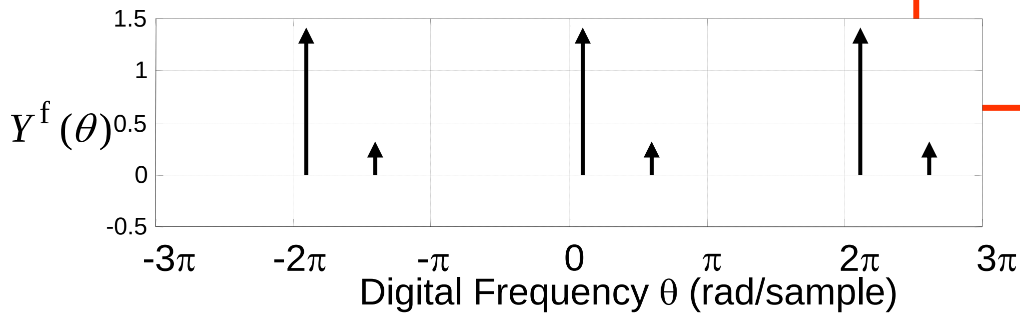
Consider a signal consisting of two complex sinusoids:

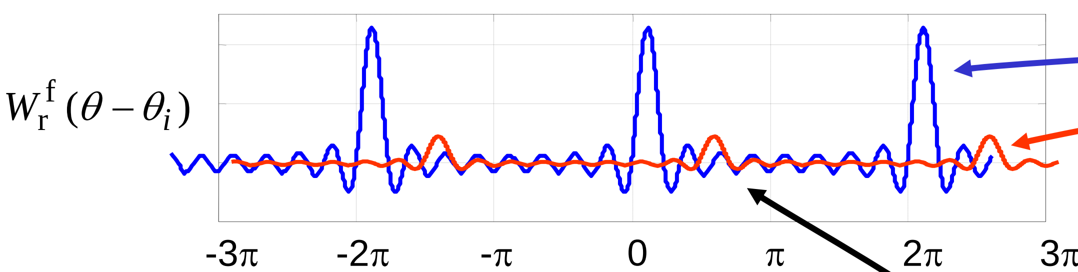
$$y[n] = A_1 e^{j\theta_1 n} + A_2 e^{j\theta_2 n}$$

$$Y^{\mathrm{f}}(\theta) = A_1\delta(\theta-\theta_1) + A_2\delta(\theta-\theta_2), \quad \theta \in [-\pi, \pi]$$ Repeats Elsewhere

Recall: $F(\theta)*\delta(\theta-\alpha) = F(\theta-\alpha)$ so….

$$X^{\mathrm{f}}(\theta) = \frac{1}{2\pi}\left\{ Y^{\mathrm{f}} \underset{N}{\circledcirc} W_r^{\mathrm{f}} \right\}(\theta) = \frac{1}{2\pi}\left[A_1 W_r^{\mathrm{f}}(\theta-\theta_1) + A_2 W_r^{\mathrm{f}}(\theta-\theta_2) \right]$$

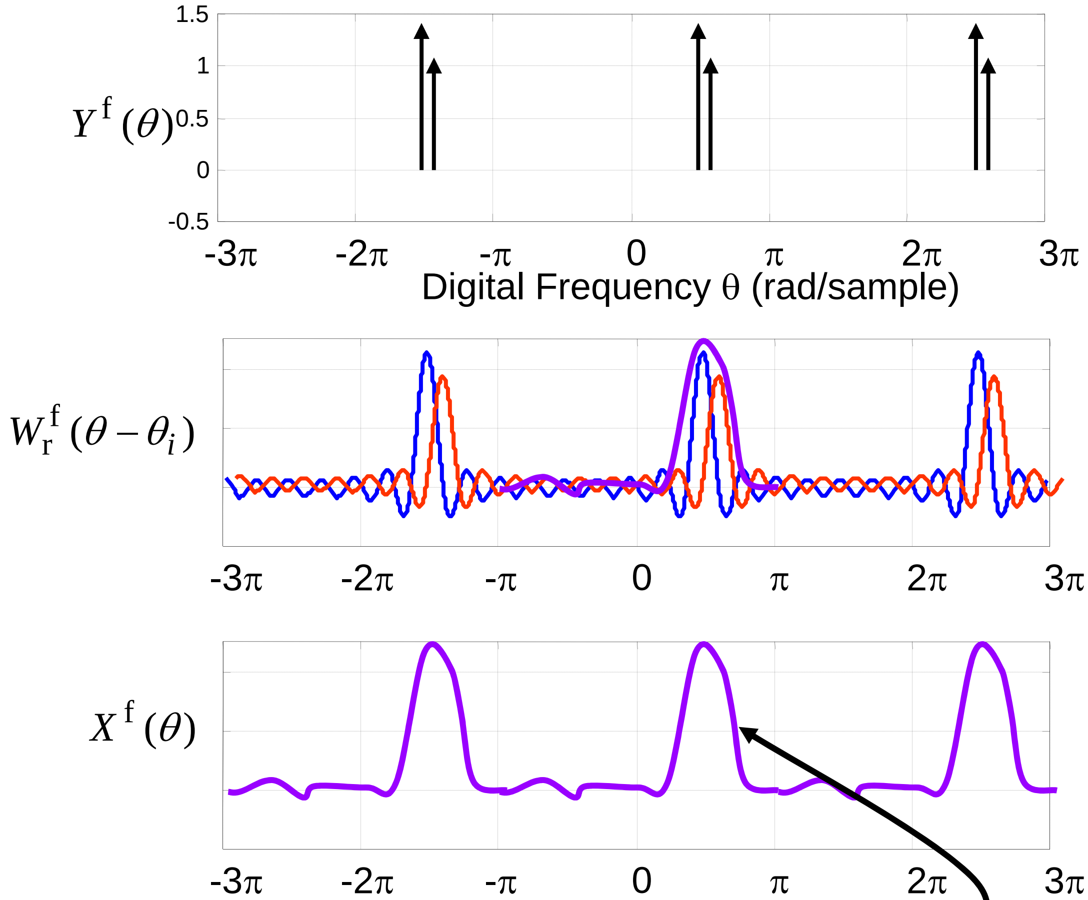




Sidelobe Leakage ("SL Interference")
Large Sidelobes Obscure Small Sinusoid

Impact of Window (pt. 3)

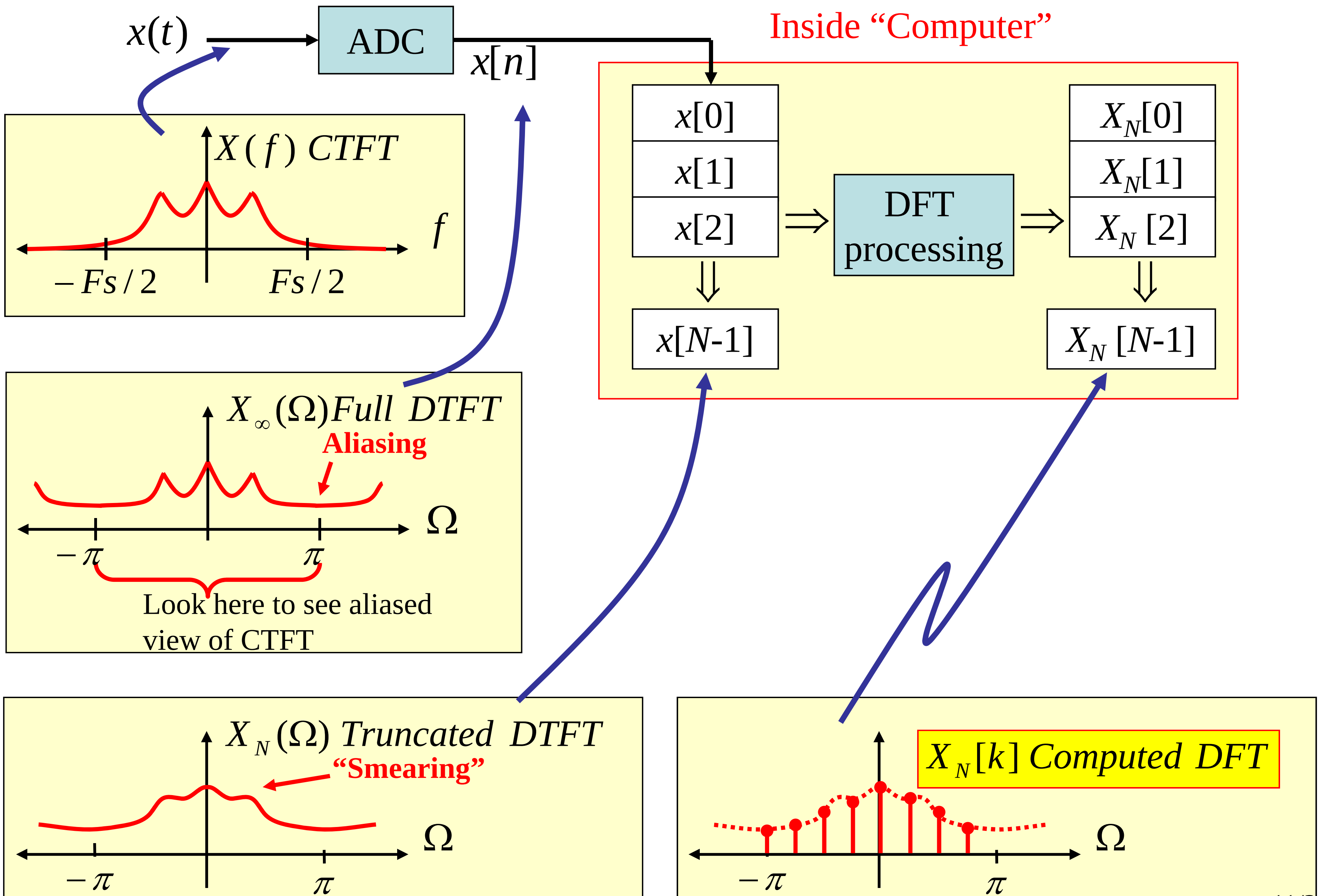
Consider a signal consisting of two complex sinusoids **closely spaced** in frequency and similar in amplitude:



Mainlobe Smearing

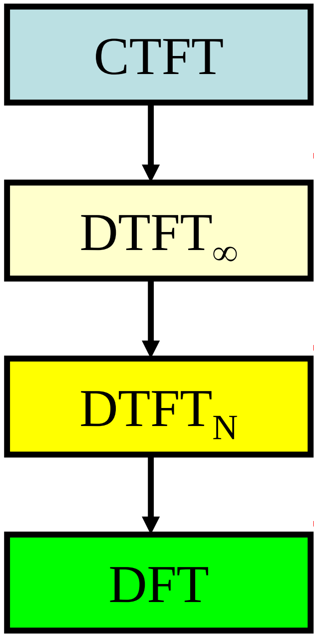
Wide Mainlobe "Smears" Sinusoids Together

NS-21 CTFT-DTFT-DFT Connections – Summary



NS-21 CTFT-DTFT-DFT Connections – Summary

Errors in a Computed DFT



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

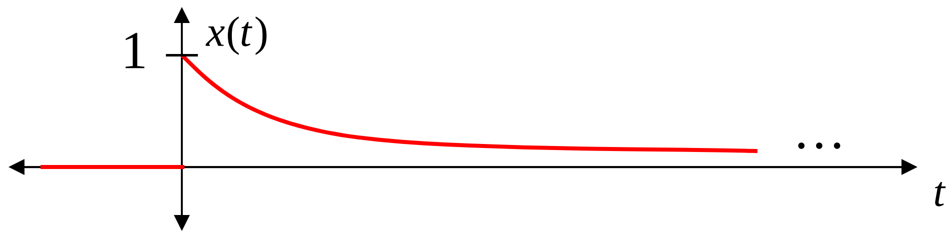
"Smearing" Error – control through { *N* choice, "window" choice }

"Grid" Error – control through { *N* choice, "zero padding" }

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!!!

NS-21 CTFT-DTFT-DFT Connections – Example

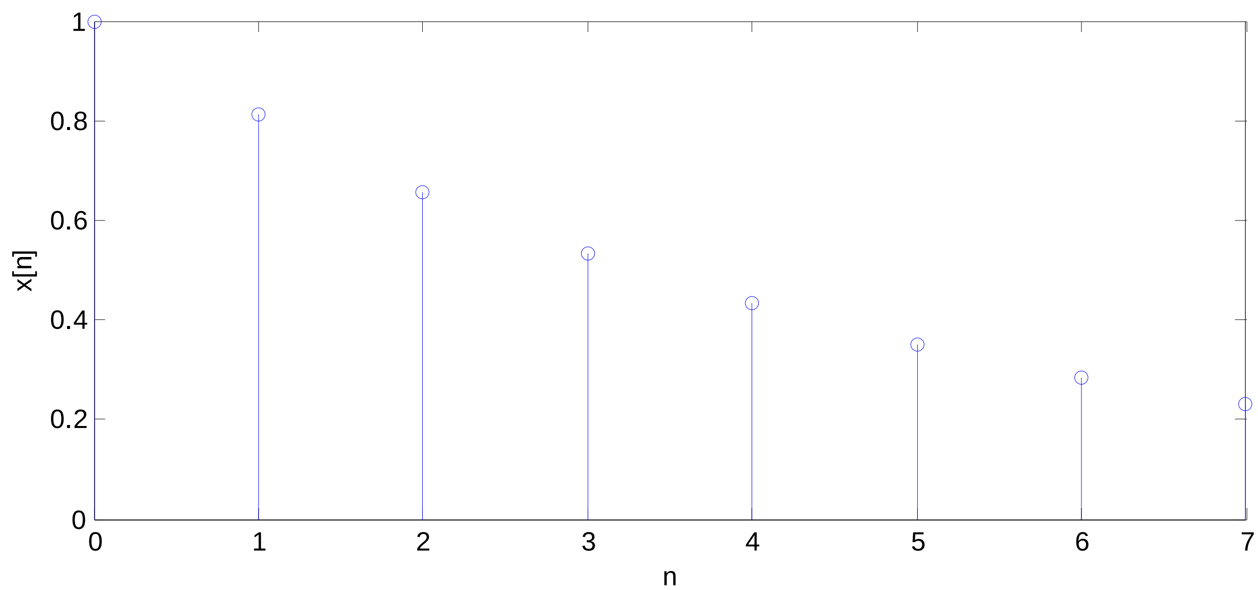
Let's imagine we have the following CT Signal: $x(t) = e^{-bt}u(t) \quad for \;\; b > 0$



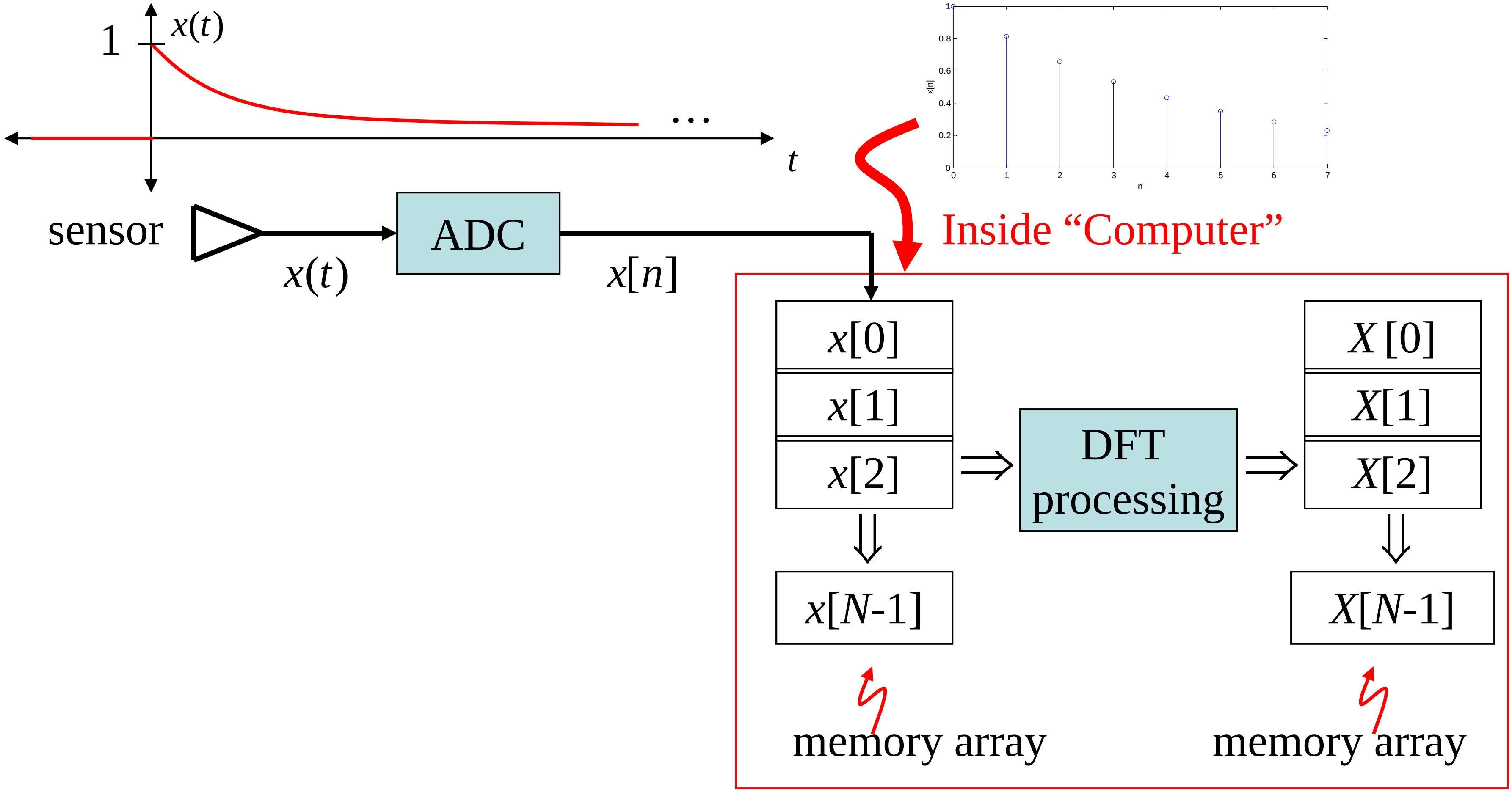
Suppose that $b = 2000\pi$

Suppose we sample this signal at F_s = 30 kHz

Suppose we take 8 samples….



Now put these 8 samples into our computer and compute the DFT without ZP:



Let's ***simulate*** doing this by using MATLAB!!

$$x(t) = e^{-2000\pi t}u(t) \qquad F_s = 30 \text{ kHz} \qquad T = 1/F_s$$

```
Fs=30000; T=1/Fs;
n=0:7;
t_n=n*T;
b=2000*pi;
x=exp(-b*t_n);
subplot(2,1,1);  stem(n,x); xlabel('n');ylabel('x[n]')

%%%% Now we have 8 signal samples of this signal
%%%  Now we compute the DFT of it using the fft command
%%%     ....and we use fftshift to put the results between -pi and pi

X_dft=fftshift(fft(x));

%%%  Now plot magnitude of DFT
N = length(X_dft);
delta_f = Fs/N;
f=(-(N/2):((N/2)-1))*delta_f;
subplot(2,1,2); stem(f,abs(X_dft)); xlabel('f (Hz)');ylabel('|DFT|')
```

We have three choices:
(a) Against index k values
(Rarely a good choice)
(b) Against DT freq Omega (rad/sample) between -pi and pi
(Use if you only care about "DT world")
(c) Against CT frequency f (Hz) between -Fs/2 and Fs/2
(Use when you care about "link to CT world")

See Next Slide…

Creation of Frequency Vector for DFT Plotting

When computing the DFT (using fft) you get N numbers that tell the *values* the DFT coefficients have. But you need to know what frequencies they are at…

We'll assume that you are using fftshift, which moves the DFT coefficients around so they lie in the frequency range -π to π

To plot versus Ω in rad/sample:

- For N DFT points… the frequency spacing between them is $2\pi/N$
- With fftshift, the frequencies start at -π
- Thus the command that makes these frequency points is

omega = ((-N/2):((N/2) -1))*2*pi/N

To plot versus f in Hz:

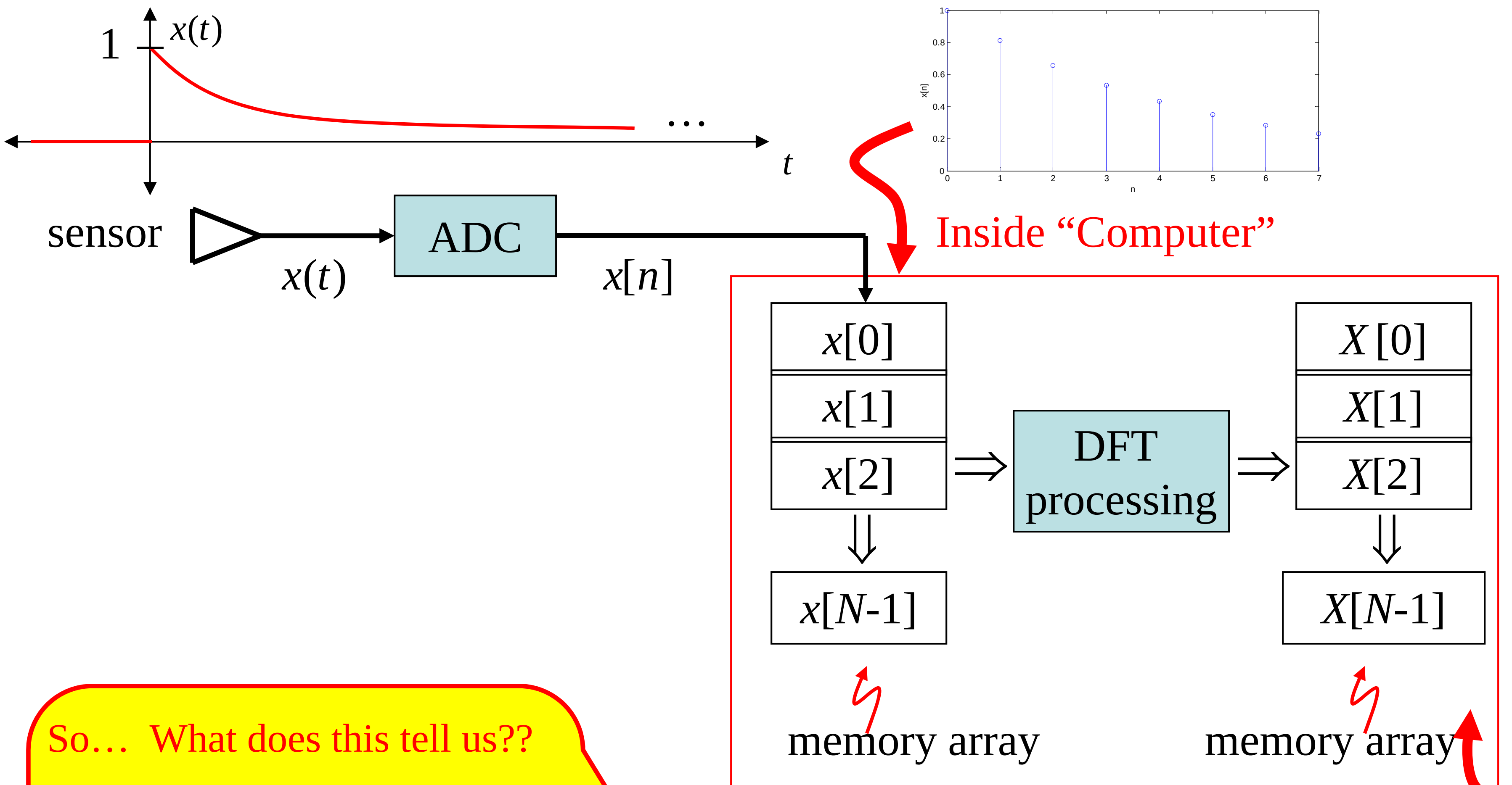
- For N DFT points… the frequency spacing between them is F_s/N
- With fftshift, the frequencies start at $-F_s/2$
- Thus the command that makes these frequency points is

f = ((-N/2):((N/2) -1))*Fs/N

8 points
Starts at pi
Stops "just shy of pi"

Example for our N=8 case: omega = (-4:3)*2*pi/8

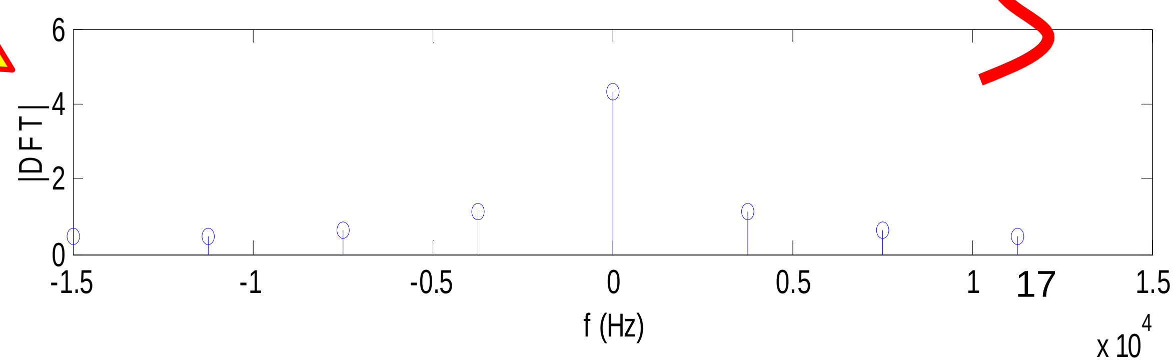
gives the vector [-pi -3pi/4 -pi/2 –pi/4 0 pi/4 pi/2 3pi/4]

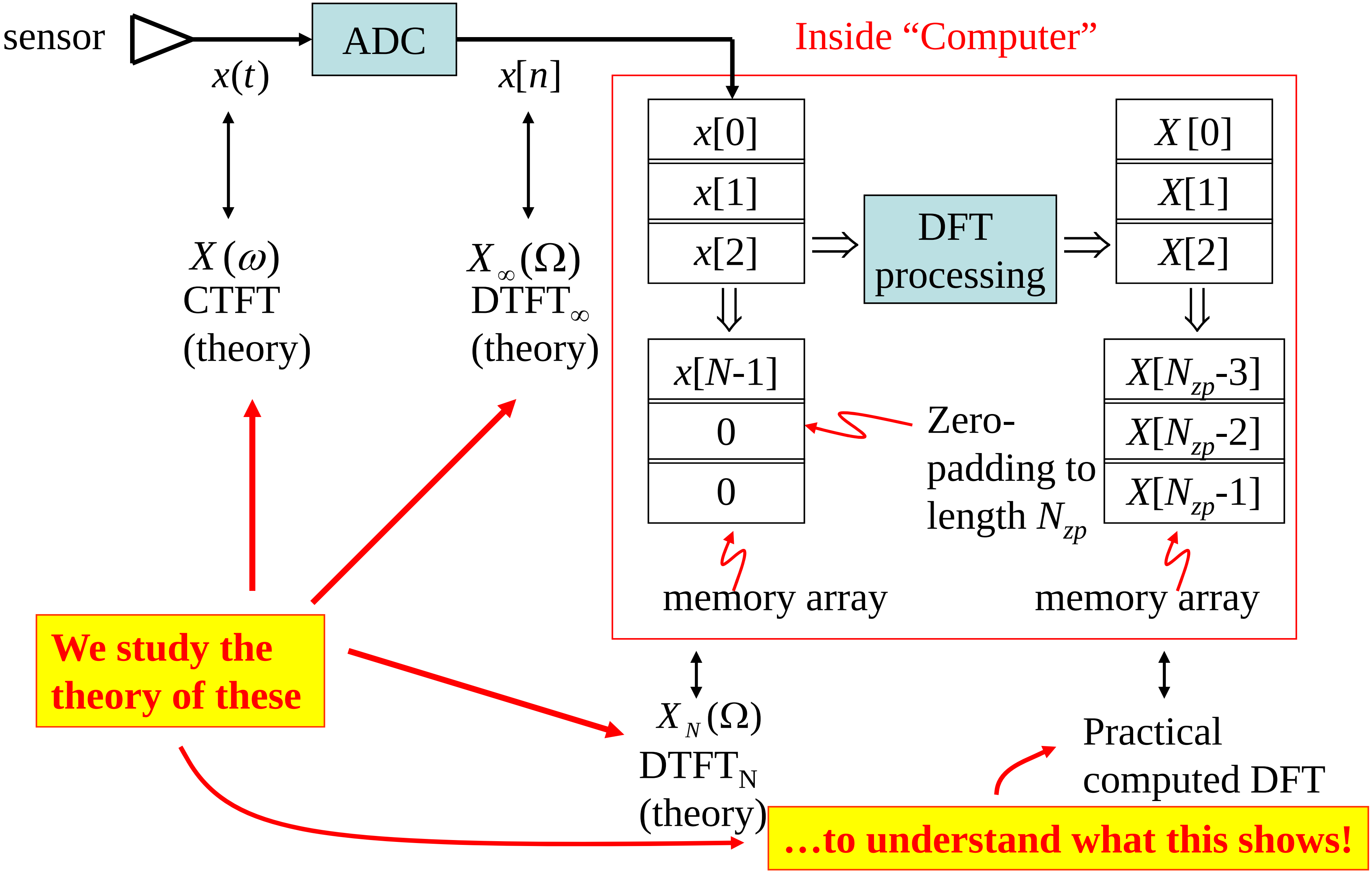


So… What does this tell us??

How does this relate to the CT signal’s CTFT?

How do we answer those questions???





So… to analyze what we get from the DFT processing for this signal…

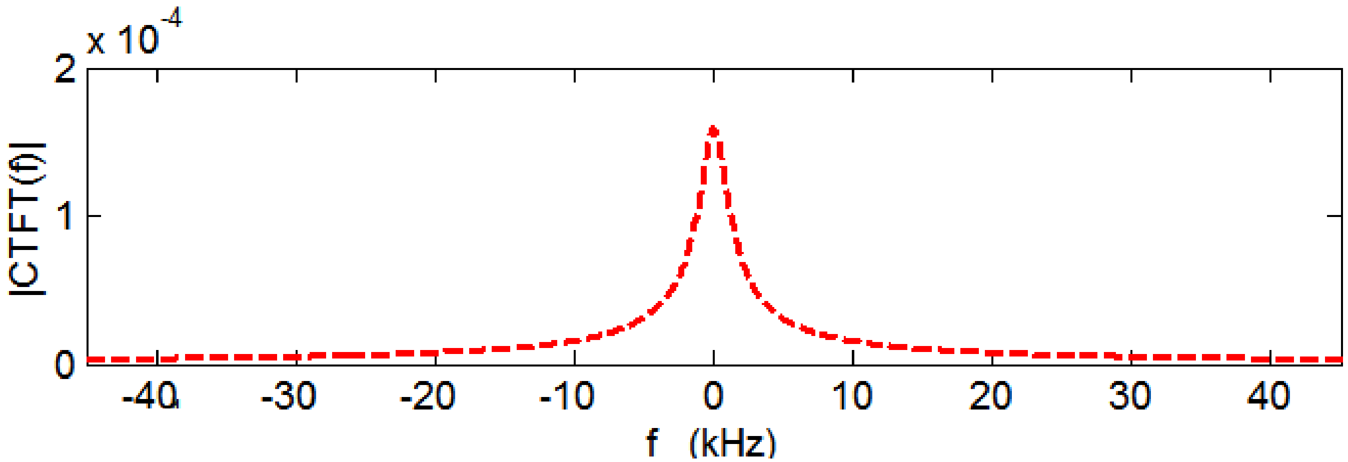
… Start at the CTFT @ ADC input!!!!

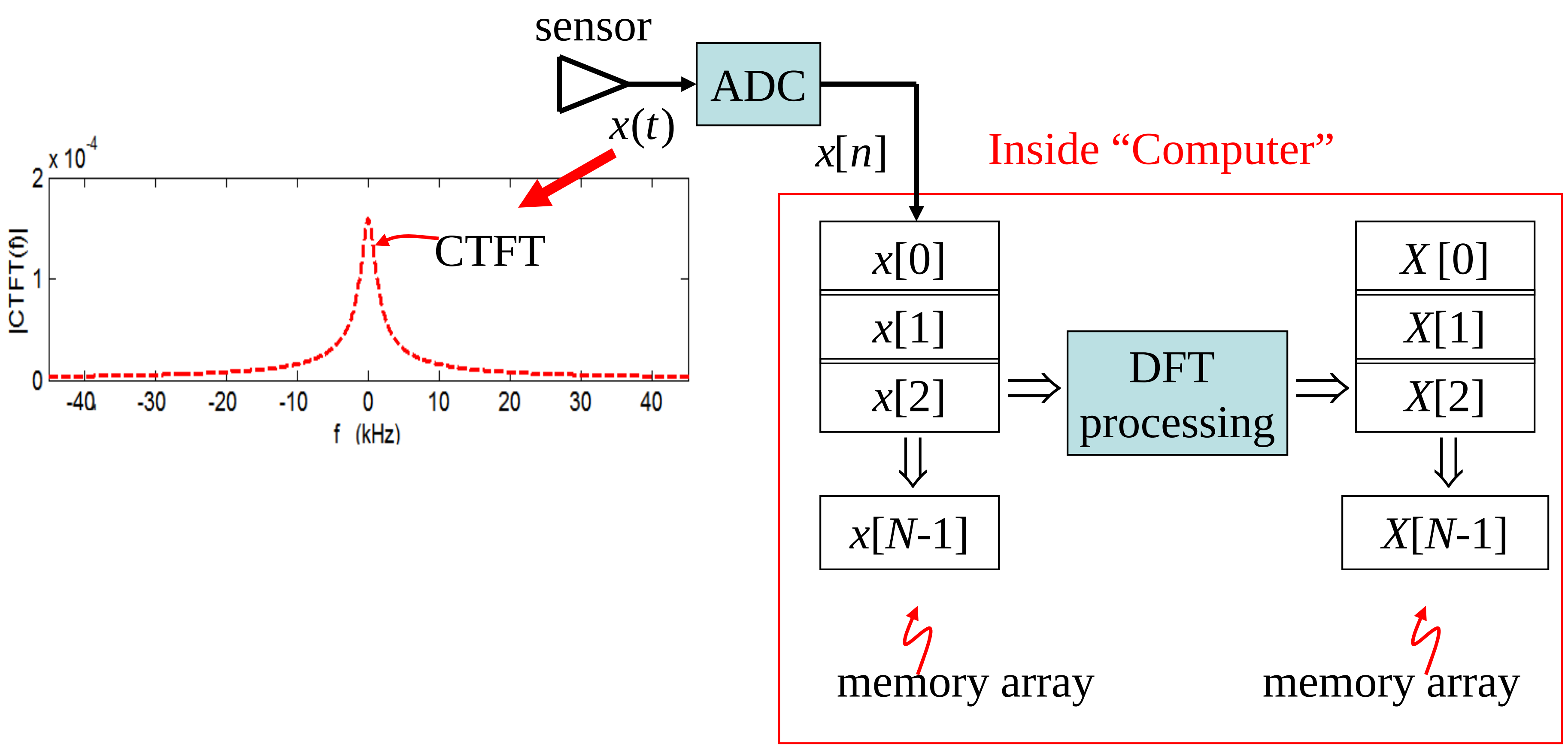
CTFT of Signal at ADC Input

$$x(t) = e^{-bt}u(t)$$

From FT Table we have: $X(\omega) = \dfrac{1}{j\omega + b} \Rightarrow X(f) = \dfrac{1}{j2\pi f + b}$

```
%%%% Compute the Theoretical CTFT
b=2*pi*1000;
f=-200000:100:200000;
CTFT=1./(j*2*pi*f + b);   %%% from CTFT table
plot(f/1e3,abs(CTFT),'r--');
```





sensor
ADC
x(t)
x[n]
Inside "Computer"
x 10^-4
2
1
0
|CTFT(f)|
CTFT
-40
-30
-20
-10
0
10
20
30
40
f (kHz)
x[0]
x[1]
x[2]
x[N-1]
DFT processing
X [0]
X[1]
X[2]
X[N-1]
memory array
memory array

DTFT of Signal at ADC Output

If we sample $x(t)$ at the rate of F_s samples/second – That is, sample every $T = 1/F_s$ sec – we get the DT Signal coming out of the ADC is:

$$x[n] = x(t)|_{t=nT} = x(nT)$$

For this example we get:

$$x[n] = \left[e^{-bt}u(t)\right]_{t=nT} = e^{-bTn}u[n]$$

Sampled Signal

$$= \left(e^{-bT}\right)^n u[n] \triangleq a^n u[n]$$

Note: $|a| < 1$

Now imagine that in theory we have all of the samples $x[n]$ $-\infty < n < \infty$ at the ADC output.

Then, in theory the DTFT_∞ of this signal is found using the DTFT table to be:

$$X_\infty(\Omega) = \frac{1}{1 - ae^{-j\Omega}}$$

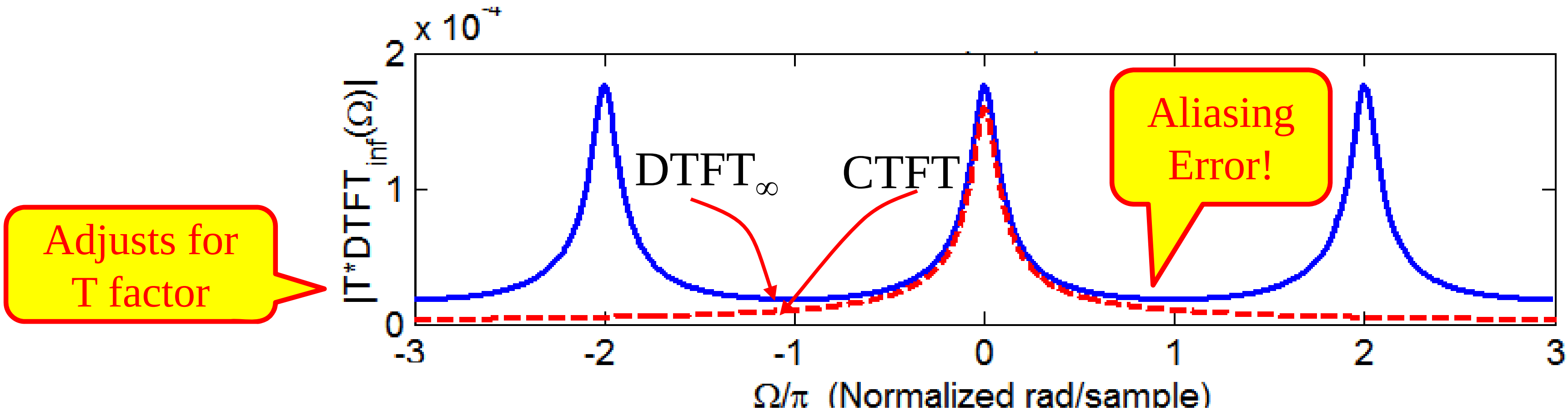
DTFT_∞ Result…(Theory)

For $|a| < 1$ which we have because:

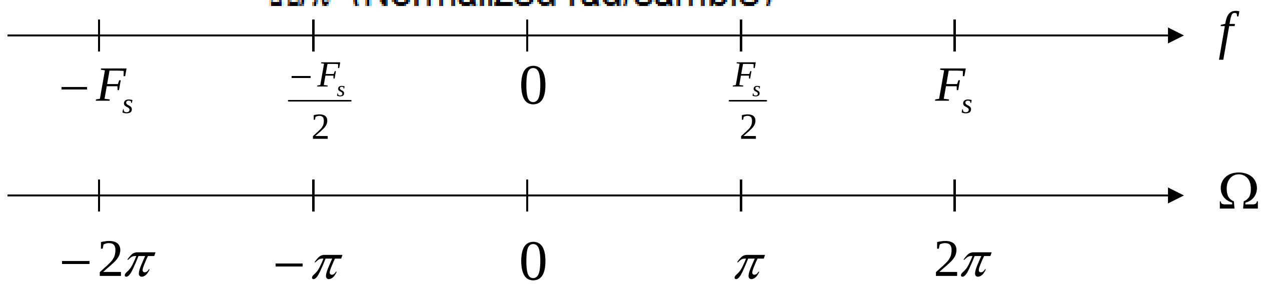
$$a = e^{-bT} \;\; \& \; b > 0, \; T > 0$$

$$X_{\infty}(\Omega) = \frac{1}{1 - ae^{-j\Omega}}$$

```
%%%%  Compute the Theoretical DTFT_inf
T=1/Fs;
a=exp(-b*T);   %%% computes exponential decay rate of sampled signal
omega=-3*pi:0.01:3*pi;
DTFT_inf=1./(1 - a*exp(-j*omega));   %%% from DTFT table
plot(omega/pi,abs(T*DTFT_inf));
xlabel('\Omega/\pi  (Normalized rad/sample)')
ylabel('|T*DTFT_{inf}(\Omega)|')
hold on
h=plot(f/(Fs/2),abs(CTFT),'r--');
axis_x([-3 3])
hold off
```



Recall the two equivalent axes:

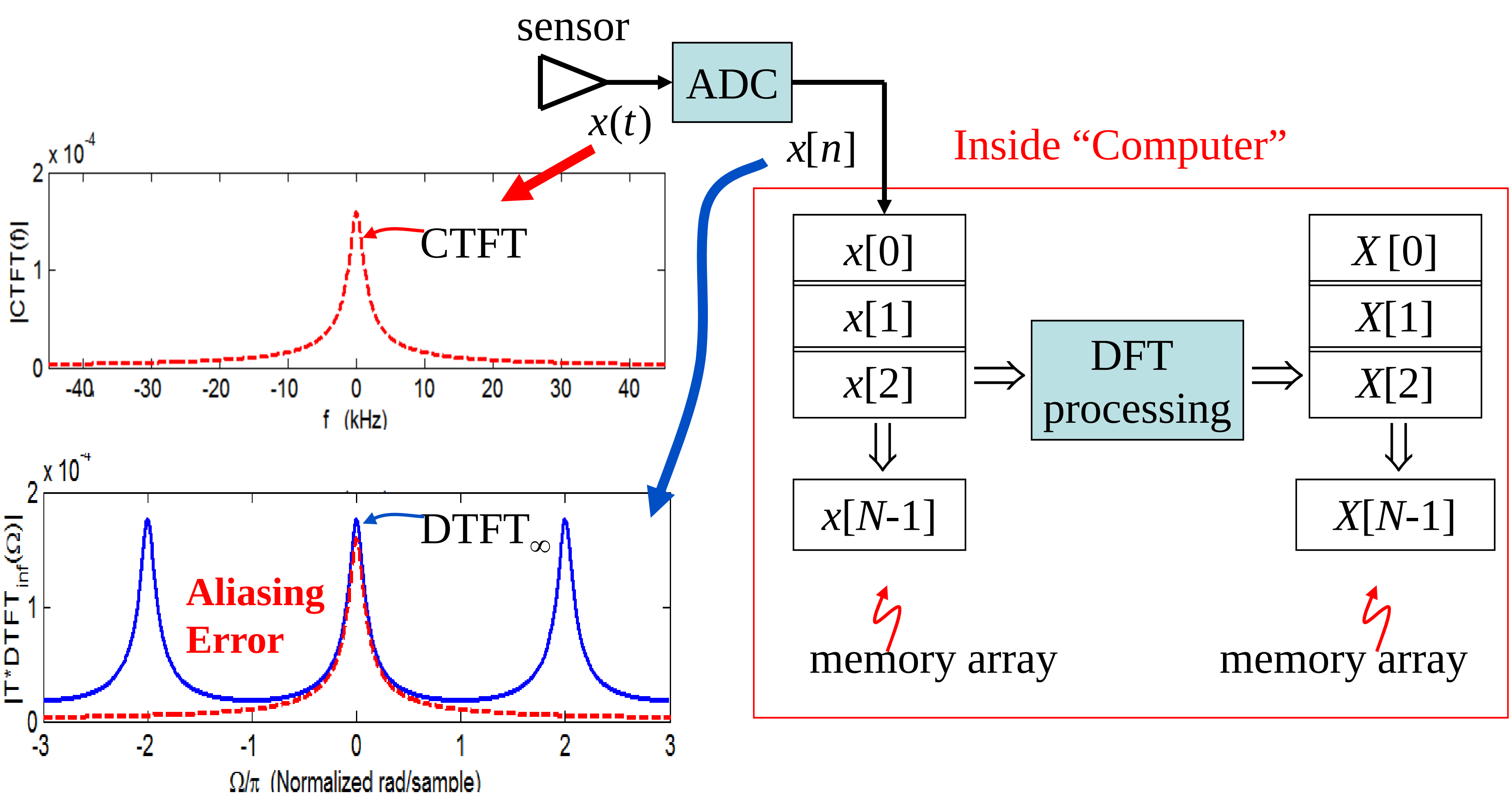


Our theory says that:

(For analysis) $$X_\infty(\Omega) = \frac{1}{T}\underbrace{\sum_{k=-\infty}^{\infty} X\left(\left(\frac{\Omega + k2\pi}{2\pi}\right)F_s\right)}$$

So we should see "replicas" in $X_\infty(\Omega)$ and we do!

We plot $TX_\infty(\Omega)$ to undo the 1/T here



sensor
ADC
x(t)
x[n]
Inside "Computer"
x 10^-4
|CTFT(f)|
CTFT
f (kHz)
x 10^-4
|T*DTFT_inf(Ω)|
DTFT∞
Aliasing Error
Ω/π (Normalized rad/sample)
x[0]
x[1]
x[2]
x[N-1]
DFT processing
X [0]
X[1]
X[2]
X[N-1]
memory array
memory array

DTFT of Signal Stored Inside Computer

Now, in reality we can "collect" only $N < \infty$ samples in our computer:

$$x_N[n] = a^n,\ 0 \le n \le (N-1)$$

$$(\text{"Assume"}\ \ x_N[n] = 0\ \ \text{elsewhere})$$

The DTFT of this collected finite-duration signal is easily found "by hand":

$$X_N(\Omega) = \frac{1-\left(ae^{-j\Omega}\right)^N}{1-ae^{-j\Omega}}$$

Eq. (7.6) in Kamen & Heck

Note that we think of this as follows:

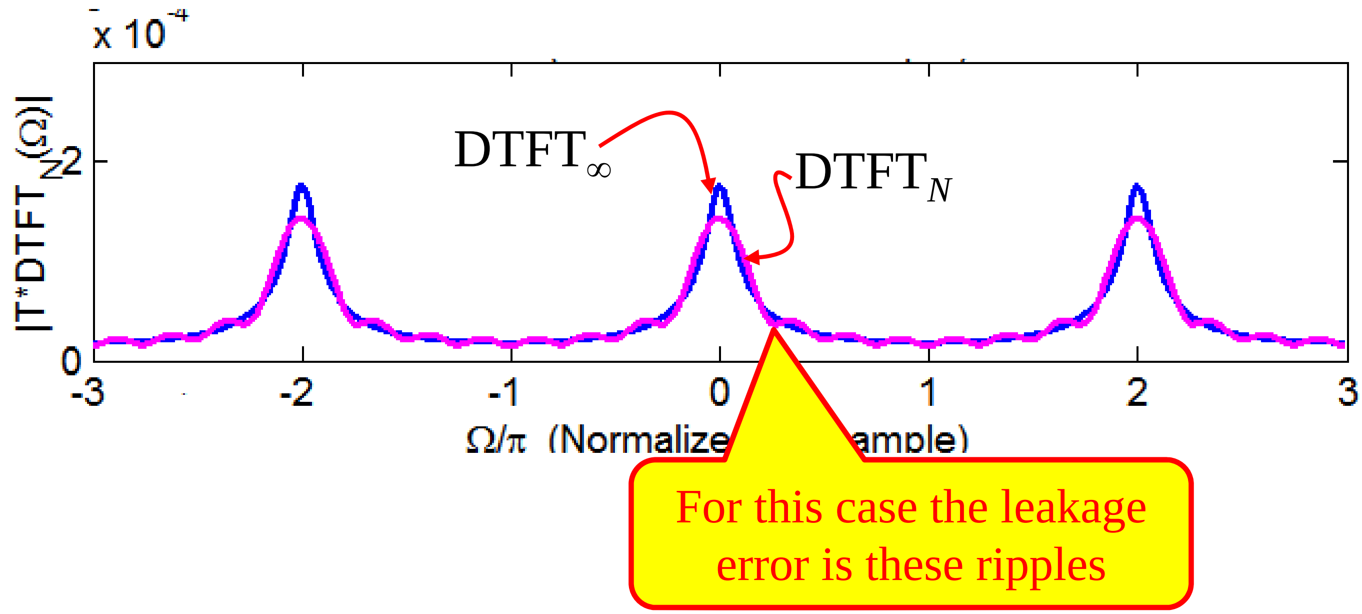
$$x_N[n] = x[n]w_N[n] \qquad w_N[n] = \begin{cases} 1, & 0, 1, 2, \ldots, N-1 \\ 0, & otherwise \end{cases}$$

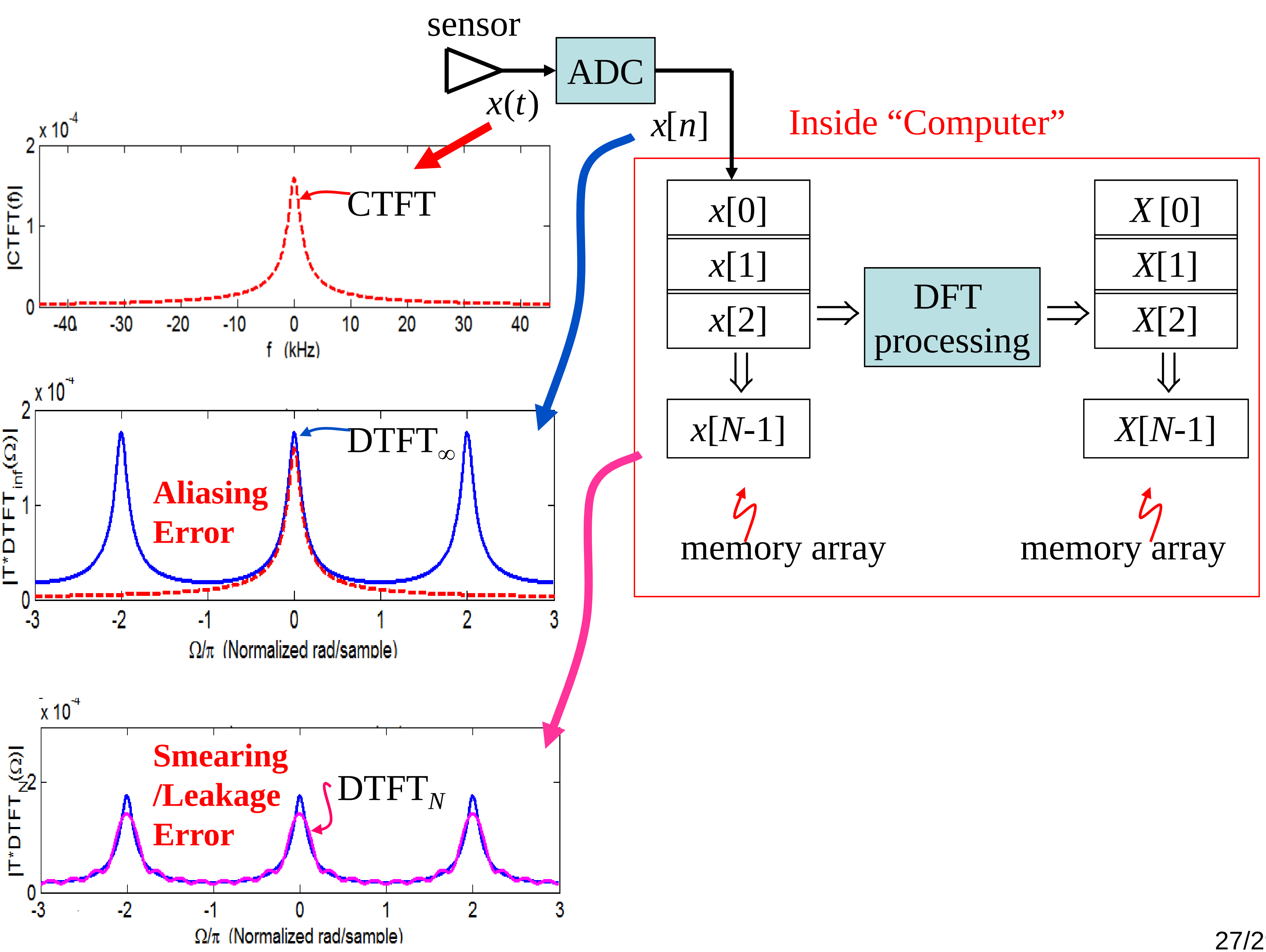
..and DTFT theory tells us that

$$X_N(\Omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X_\infty(\Omega-\lambda)W_N(\lambda)d\lambda$$

…and this has a "smearing" effect: called "leakage error"

```
%%%%  Compute the Theoretical DTFT_N
DTFT_N=(1-(a*exp(-j*omega)).^N)./(1 - a*exp(-j*omega));
plot(omega/pi,abs(T*DTFT_inf));
hold on
plot(omega/pi,abs(T*DTFT_N),'m');
hold off
xlabel('\Omega/\pi  (Normalized rad/sample)')
ylabel('|T*DTFT_{N}(\Omega)|')
```




sensor
ADC
x(t)
x[n]
Inside "Computer"
x[0]
x[1]
x[2]
x[N-1]
DFT processing
X[0]
X[1]
X[2]
X[N-1]
memory array
memory array
CTFT
|CTFT(f)|
f (kHz)
DTFT∞
Aliasing Error
|T*DTFT_inf(Ω)|
Ω/π (Normalized rad/sample)
Smearing /Leakage Error
DTFT_N
|T*DTFT_N(Ω)|
Ω/π (Normalized rad/sample)

Finally… DFT of Signal Stored Inside Computer

$$x_N[n] = \begin{bmatrix} x[0] \\ x[1] \\ \dots \\ x[N-1] \end{bmatrix}$$

$$X_N[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi kn}{N}}$$

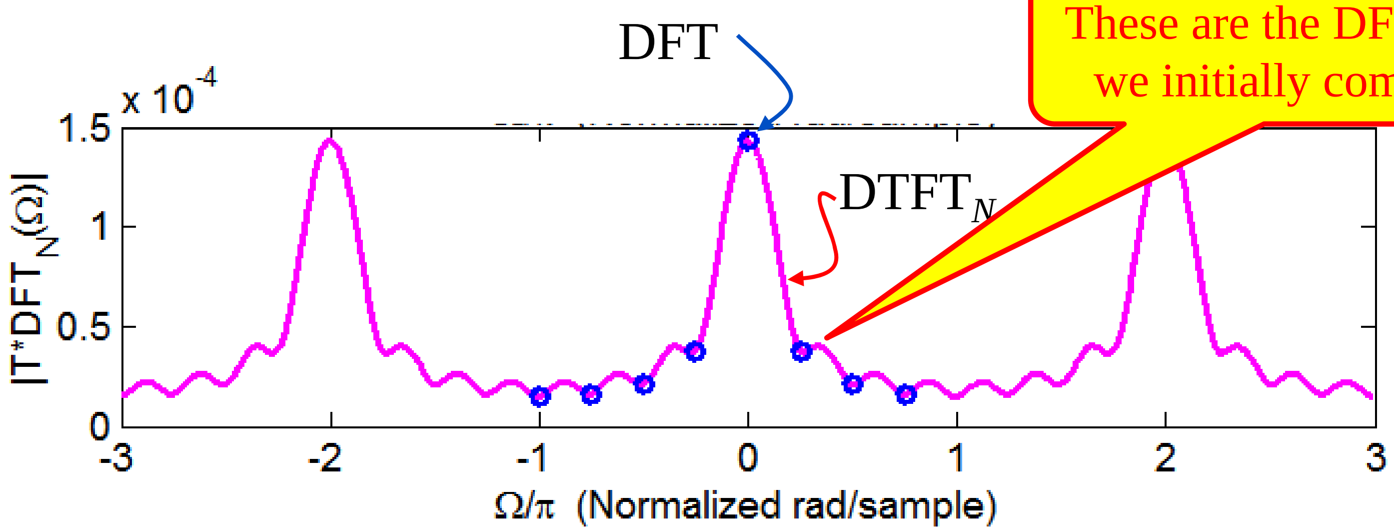
(The only part of this example we'd <u>really</u> "do")

Our theory tells us that the zero-padded DFT is nothing more than "points" on DTFT$_N$:

$$X_{zp}[k] = X_N(\Omega_k)$$

where $\Omega_k = \frac{2\pi k}{N}$ $k = 0, 1, 2, \ldots, N-1$

Spacing between DFT "points"



Summary of Results: