BINGHAMTON UNIVERSITY

State University of New York

EECE 301
Signals & Systems
Prof. Mark Fowler

Note Set #8

- C-T Signals: Computing the FS Coefficients

Analytically Finding FS Coefficients

Q: How do we find the Exponential Form FS Coefficients?

A: Use this: (it can be proved but we won't do that here!)

$$c_k = \frac{1}{T}\int_{t_0}^{t_0+T} x(t)e^{-jk\omega_0 t}\,dt$$

Integrate over any complete period

Some books use only $t_0 = 0$.

where: T = fundamental period of $x(t)$ (in seconds)

ω_0 = fundamental frequency of $x(t)$ (in rad/second)

$= 2\pi/T$

t_0 = any time point (you pick t_0 to ease calculations)

$k \in$ all integers (… –3, –2, –1, 0, 1, 2, 3, …)

Looks like we have to do this integral infinitely many times!!! But…Usually you can do the integral in terms of arbitrary *k*!

Comment: Note that for $k = 0$ this gives

$$c_0 = \frac{1}{T}\int_{t_0}^{t_0+T} x(t)\,dt$$

c_0 is the "DC offset", which is the time-average over one period

Q: How do we find the Sine-Cosine Form FS Coefficients?

A: Use these: (can be proved but we won't do that here!)

$$a_0 = \frac{1}{T}\int_{t_0}^{t_0+T} x(t)dt$$

a_0 is the "DC offset", which is the time-average over one period

$$a_k = \frac{2}{T}\int_{t_0}^{t_0+T} x(t)\cos\left(k\omega_0 t\right)dt$$

$$b_k = \frac{2}{T}\int_{t_0}^{t_0+T} x(t)\sin\left(k\omega_0 t\right)dt$$

Integrate over any complete period

where: T = fundamental period of $x(t)$ (in seconds)

ω_0 = fundamental frequency of $x(t)$ (in rad/second)

$= 2\pi/T$

t_0 = any time point (you pick t_0 to ease calculations)

$k \in$ all integers

Q: How do we find the Amplitude-Phase Form FS Coefficients?

A: No easy direct way! So convert from one of the other forms!

$$A_0 = a_0$$

$$A_k = \sqrt{a_k^2 + b_k^2}$$

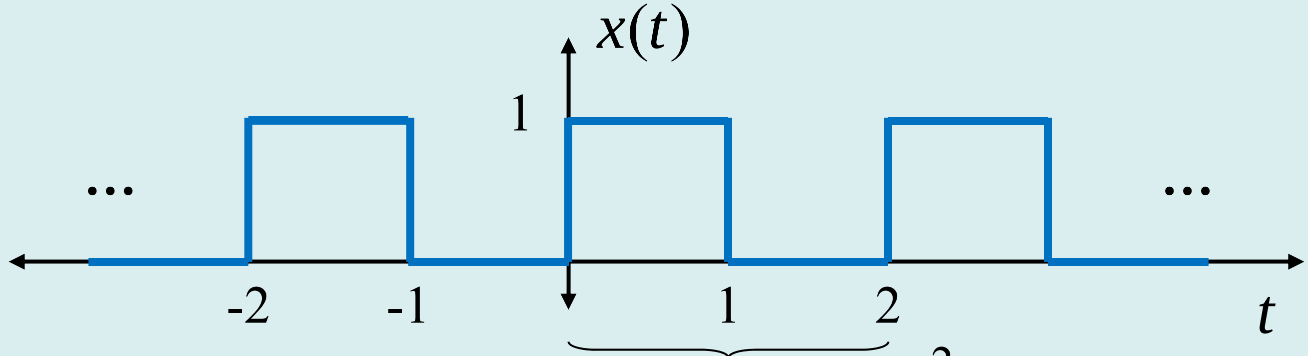
$$\theta_k = \tan^{-1}\left(\frac{-b_k}{a_k}\right)$$

$$A_0 = c_0$$

$$\left.\begin{aligned} A_k &= 2|c_k| \\ \theta_k &= \angle c_k \end{aligned}\right\} k = 1, 2, 3, \ldots$$

- Recall… you can convert from any form into any other form using some simple equations!
- Thus… I tend to always find the c_k and then convert to other forms if needed.
- Why do I prefer to find the c_k?
  - Only one integral to actually do (although it is complex valued!)
  - Integrals involving exponential are usually easier than for sinusoids!

<u>Example</u>: FS of Rectangular Pulse Train



$$T = 2 \;\Rightarrow\; \omega_0 = \frac{2\pi}{2} = \pi \text{ rad/sec}$$

$$c_k = \frac{1}{T}\int_{t_0}^{t_0+T} x(t)e^{-jk\omega_0 t}dt$$

choose $t_0 = 0$

$$= \frac{1}{2}\int_0^2 x(t)e^{-jk\pi t}dt$$

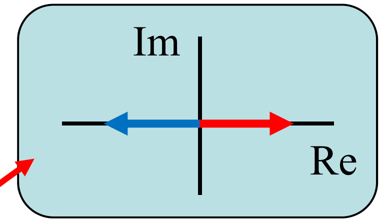
$$= \frac{1}{2}\left[\int_0^1 1e^{-jk\pi t}dt + \int_1^2 0 \times e^{-jk\pi t}dt\right]$$

$$= \frac{1}{2}\int_0^1 e^{-jk\pi t}dt$$

$$= \frac{1}{2}\left[\frac{1}{-jk\pi}e^{-jk\pi t}\right]_0^1$$

$$= \frac{j}{2(k\pi)}\left[e^{-jk\pi} - 1\right]$$

$$e^{-jk\pi} = \begin{cases} 1, & k \text{ even} \\ -1, & k \text{ odd} \end{cases}$$



$$c_k = \begin{cases} 0, & k \text{ even}, \neq 0 \\ \dfrac{-j}{k\pi}, & k \text{ odd} \end{cases}$$

Not valid for $k = 0$… so have to do that case separately!

$$c_0 = \frac{1}{2}\int_0^1 1e^{-j0\pi t}dt = \frac{1}{2}\int_0^1 1dt$$

$$c_0 = \frac{1}{2}$$

DC Level (also called DC Offset)

So… we've found the exponential FS to be:

$$x(t) = \cdots + \tfrac{-j}{-3\pi} e^{-j3\omega_o t} + \tfrac{-j}{-1\pi} e^{-j1\omega_o t} + \tfrac{1}{2} + \tfrac{-j}{1\pi} e^{j1\omega_o t} + \tfrac{-j}{3\pi} e^{j3\omega_o t} + \cdots$$

$$c_k = \begin{cases} \frac{1}{2}, & k = 0 \\ 0, & k \text{ even}, \neq 0 \\ \frac{-j}{k\pi}, & k \text{ odd} \end{cases}$$

$$a_0 = c_0$$

$$a_k = 2\,\mathrm{Re}\{c_k\}, \quad k = 1, 2, 3, \ldots$$

$$b_k = -2\,\mathrm{Im}\{c_k\}, \quad k = 1, 2, 3, \ldots$$

$$a_k = \begin{cases} \frac{1}{2}, & k = 0 \\ 0, & k \neq 0 \end{cases}$$

$$b_k = \begin{cases} 0, & k \text{ even} \\ \frac{2}{k\pi}, & k \text{ odd} \end{cases}$$

$$x(t) = \tfrac{1}{2} + \tfrac{2}{1\pi}\sin(1\omega_o t) + \tfrac{2}{3\pi}\sin(3\omega_o t) + \tfrac{2}{5\pi}\sin(5\omega_o t) + \cdots$$

So… we've found the exponential FS to be:

$$x(t) = \cdots + \tfrac{-j}{-3\pi} e^{-j3\omega_o t} + \tfrac{-j}{-1\pi} e^{-j1\omega_o t} + \tfrac{1}{2} + \tfrac{-j}{1\pi} e^{j1\omega_o t} + \tfrac{-j}{3\pi} e^{j3\omega_o t} + \cdots$$

$$c_k = \begin{cases} \frac{1}{2}, & k = 0 \\ 0, & k \text{ even}, \neq 0 \\ \frac{-j}{k\pi}, & k \text{ odd} \end{cases}$$

$$A_0 = c_0$$

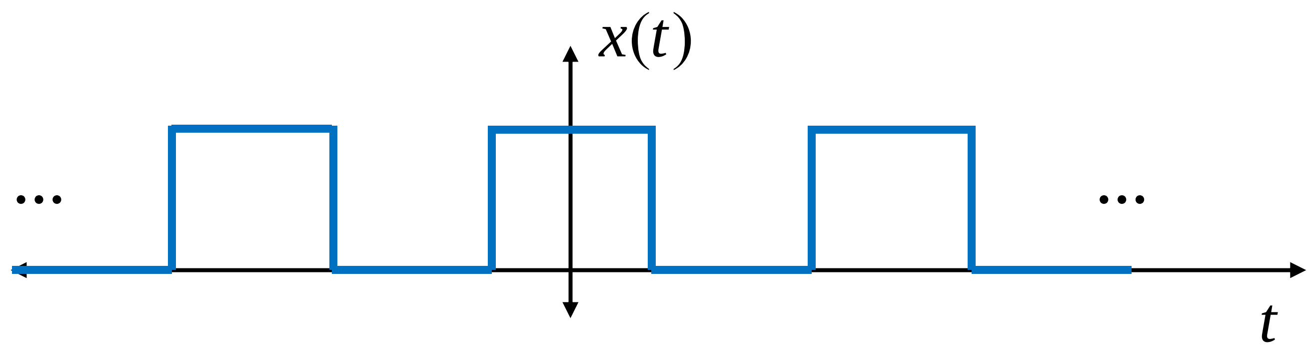
$$\left.\begin{aligned} A_k &= 2|c_k| \\ \theta_k &= \angle c_k \end{aligned}\right\} k = 1, 2, 3, \ldots$$

$$A_k = \begin{cases} \frac{1}{2}, & k = 0 \\ 0, & k \text{ even} \\ \frac{2}{k\pi}, & k \text{ odd} \end{cases} \qquad \theta_k = \begin{cases} \text{N/A}, & k = 0 \\ \text{N/A}, & k \text{ even} \\ -\frac{\pi}{2}, & k \text{ odd} \end{cases}$$

$$x(t) = \tfrac{1}{2} + \tfrac{2}{1\pi}\cos(1\omega_o t - \pi/2) + \tfrac{2}{3\pi}\cos(3\omega_o t - \pi/2) + \tfrac{2}{5\pi}\cos(5\omega_o t - \pi/2) + \cdots$$

Symmetry "Tricks" for Finding FS Coefficients

Even Symmetry: $x(-t) = x(t)$ ("flipping" around $t = 0$ does nothing)



Noting that cosines have even symmetry and sines have odd symmetry it is not surprising that an even $x(t)$ needs only cosine components in the Sine-Cosine Form:

Sine-Cosine Form	***Exp. Form***	***Amp.-Phase Form***
→ $b_k = 0$ →	$c_k =$ Are Real →	$\theta_k = \pm\pi$

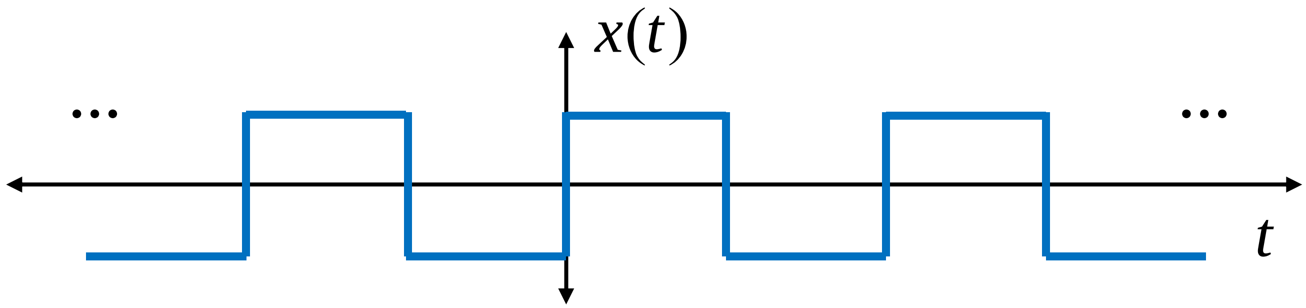
$$c_0 = a_0$$

$$\left.\begin{array}{l} c_k = \tfrac{1}{2}(a_k - jb_k) \\ c_{-k} = \tfrac{1}{2}(a_k + jb_k) \end{array}\right\} k = 1, 2, 3, \ldots$$

$$A_0 = c_0$$

$$\left.\begin{array}{l} A_k = 2|c_k| \\ \theta_k = \angle c_k \end{array}\right\} k = 1, 2, 3, \ldots$$

Odd Symmetry: $x(-t) = -x(t)$ ("flipping" around $t = 0$ negates $x(t)$)



Noting that cosines have even symmetry and sines have odd symmetry it is not surprising that an ODD $x(t)$ needs only sine components in the Sine-Cosine Form:

Sine-Cosine Form → $\underset{k\neq 0}{a_k = 0}$

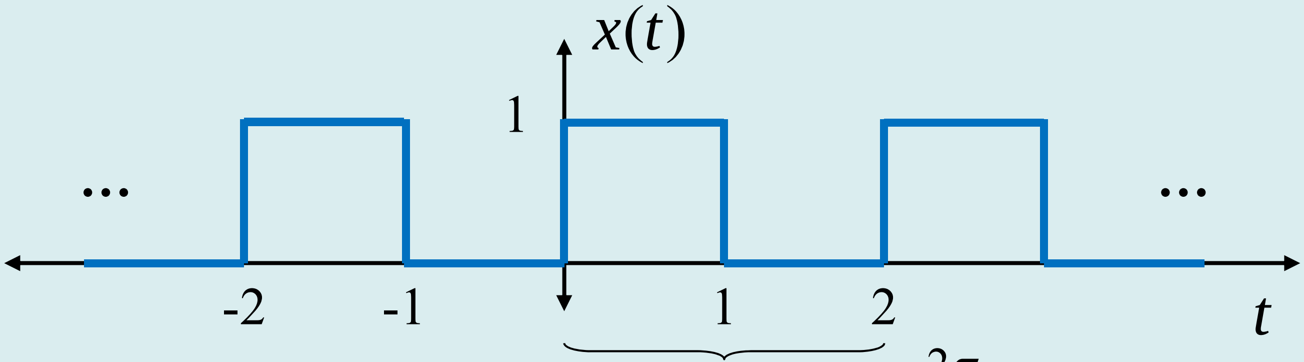
Exp. Form → $\underset{k\neq 0}{c_k = \text{ Are Imag}}$

Amp.-Phase Form → $\underset{k\neq 0}{\theta_k = \pm\pi/2}$

$$c_0 = a_0$$
$$\left.\begin{array}{l} c_k = \tfrac{1}{2}(a_k - jb_k) \\ c_{-k} = \tfrac{1}{2}(a_k + jb_k) \end{array}\right\} k = 1, 2, 3, \ldots$$

$$A_0 = c_0$$
$$\left.\begin{array}{l} A_k = 2|c_k| \\ \theta_k = \angle c_k \end{array}\right\} k = 1, 2, 3, \ldots$$

<u>Recall Example</u>: FS of Rectangular Pulse Train



$$T = 2 \;\Rightarrow\; \omega_0 = \frac{2\pi}{2} = \pi \text{ rad/sec}$$

Sine-Cosine Form	***Exp. Form***	***Amp.-Phase Form***
$a_k = 0$, $k \neq 0$	$c_k =$ Are Imag, $k \neq 0$	$\theta_k = \pm\pi/2$, $k \neq 0$

$$a_k = \begin{cases} \frac{1}{2}, & k = 0 \\ 0, & k \neq 0 \end{cases}$$

$$b_k = \begin{cases} 0, & k \text{ even} \\ \frac{2}{k\pi}, & k \text{ odd} \end{cases}$$

$$c_k = \begin{cases} 0, & k \text{ even}, \neq 0 \\ \frac{-j}{k\pi}, & k \text{ odd} \end{cases}$$

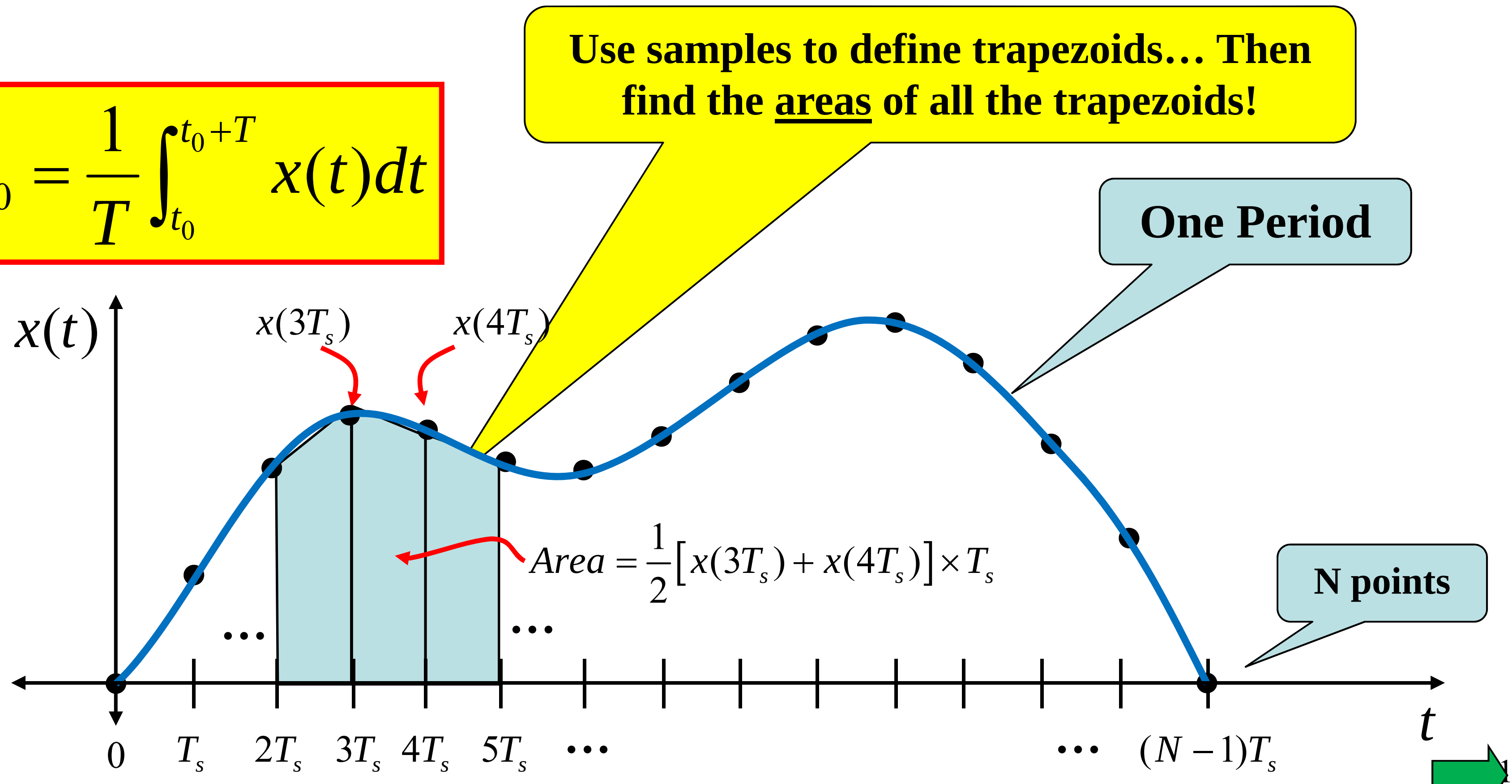
$$\theta_k = \begin{cases} \text{N/A}, & k = 0 \\ \text{N/A}, & k \text{ even} \\ -\frac{\pi}{2}, & k \text{ odd} \end{cases}$$

Numerically Finding FS Coefficients

Suppose you have a periodic signal and you want to find the FS coefficients… BUT it does not have a nice mathematical function that defines it (or it does but it is hard or impossible to do the integral)?

- **We can numerically compute the integral!**
- **Remember that an integral finds the area under a curve…**

$$c_0 = \frac{1}{T}\int_{t_0}^{t_0+T} x(t)dt$$



So… we can use samples of the integrand to compute all the trapezoid areas and then use those to approximate the integral.

Fortunately, MATLAB has a command called "trapz" that does just this!

$$c_k = \frac{1}{T}\int_{t_0}^{t_0+T} x(t)e^{-jk\omega_0 t}dt$$

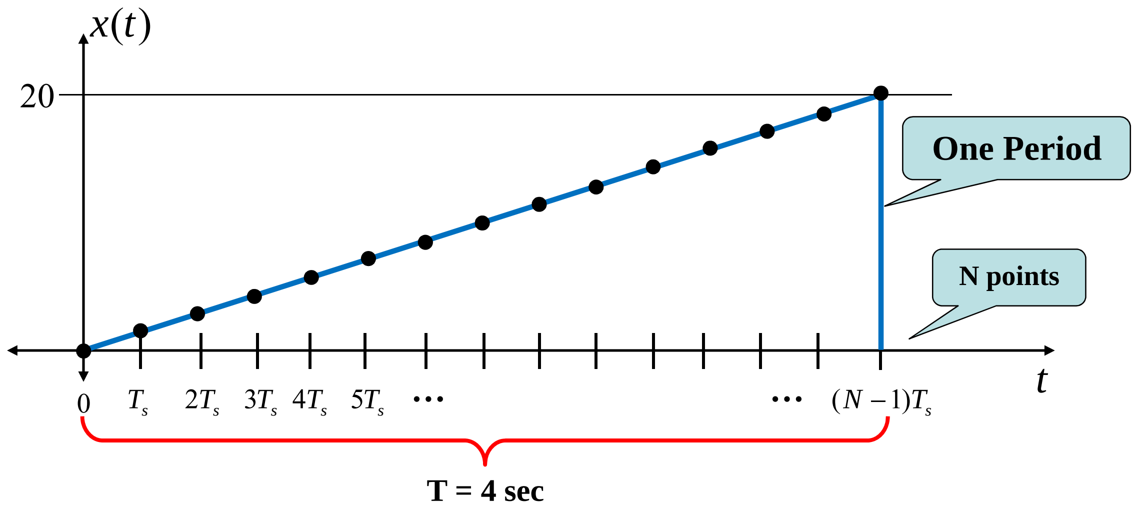
"dot star" is needed for point-by-point multiply

```
>> c_k  = (1/T)*trapz(x.*exp(-j*k*wo*t))*Ts
```

trapz assumes "unit spacing"
… so you need this

x is the vector that holds the signal samples over one period
t is a vector that holds the time values spaced Ts seconds apart
T is the period of the signal
wo is the fundamental frequency in rad/sec

Example



On the command line:

```
>>  T = 4;
>>  wo = 2*pi/T;
>> Ts = 0.2;
>>  t = 0:0.2:T;
>>  x =(20/T)*t;
>>  c_0 = (1/T)*trapz(x.*exp(-j*0*wo*t))*Ts;
>>  c_1 = (1/T)*trapz(x.*exp(-j*1*wo*t))*Ts;
>>  c_2 = (1/T)*trapz(x.*exp(-j*2*wo*t))*Ts;
```

Etc…

Stored in an m-file script:

```
T = 4;             % Specify period in seconds
wo = 2*pi/T;   % Compute fund. freq. in rad/sec
K = 10;           % specify largest k value
Ts = 0.05;         % Specify sample spacing
t = 0:Ts:T;     % Compute vector of time samples
x =(20/T)*t;    % Compute vector of signal samples
for k = (-K):K     % loop through "all" coefficients
       c(k+K+1) = (1/T)*trapz(x.*exp(-j*k*wo*t))*Ts;
end
```

Important Issues:

- How to choose sampling interval T_s?
- How to set largest *k* value??

These Are Related!!

Choosing the Sampling Interval: T_s

Once we set K (the largest k value) the FS we can compute is truncated

$$x(t) \approx \sum_{k=-K}^{K} c_k e^{jk2\pi f_0 t}$$

$$\omega_o = 2\pi f_0$$
$$\Rightarrow f_0 = 1/T$$

So the highest frequency (in Hz) is Kf_o

…so to avoid aliasing we need sampling frequency $F_s > 2Kf_o$

```
T = 4;           % Specify period in seconds
wo = 2*pi/T;     % Compute fund. freq. in rad/sec
fo = 1/T;        % Compute fund. freq. in Hz
K = 10;          % specify largest k value
Fs = 4*K*fo;     % Compute sampling rate  (set here to twice the minimum value of 2Kfo)
Ts = 1/Fs;       % Compute sample spacing
t = 0:Ts:T;      % Compute vector of time samples
x =(20/T)*t;     % Compute vector of signal samples
for k = (-K):K   % loop through "all" coefficients
     c(k+K+1) = (1/T)*trapz(x.*exp(-j*k*wo*t))*Ts;
end
```

Ends up being 0.1 rather than 0.05 as specified above!

Important Issue Remains:

- How to set largest k value??

We'll address this next and also in the next set of notes

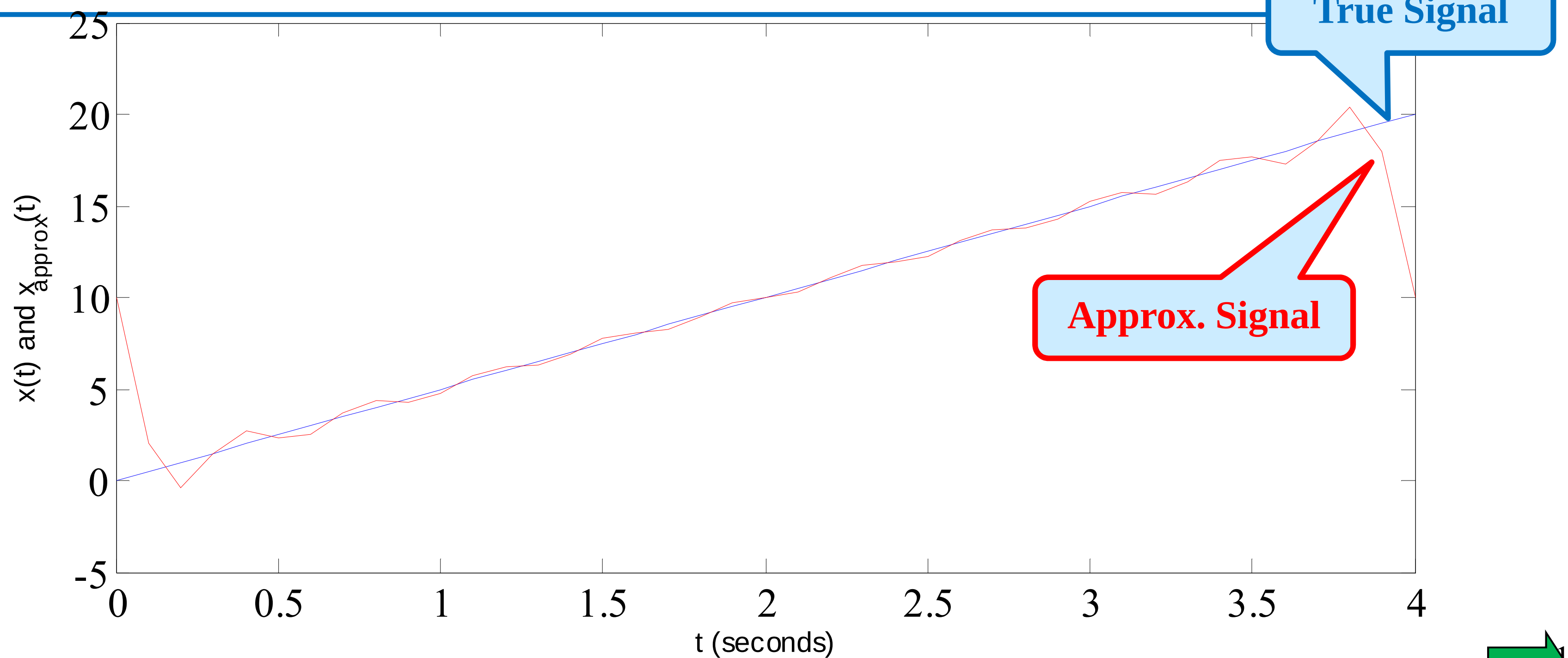
Computing the Approximate Signal

Once we have the FS coefficients… can compute the truncated series:

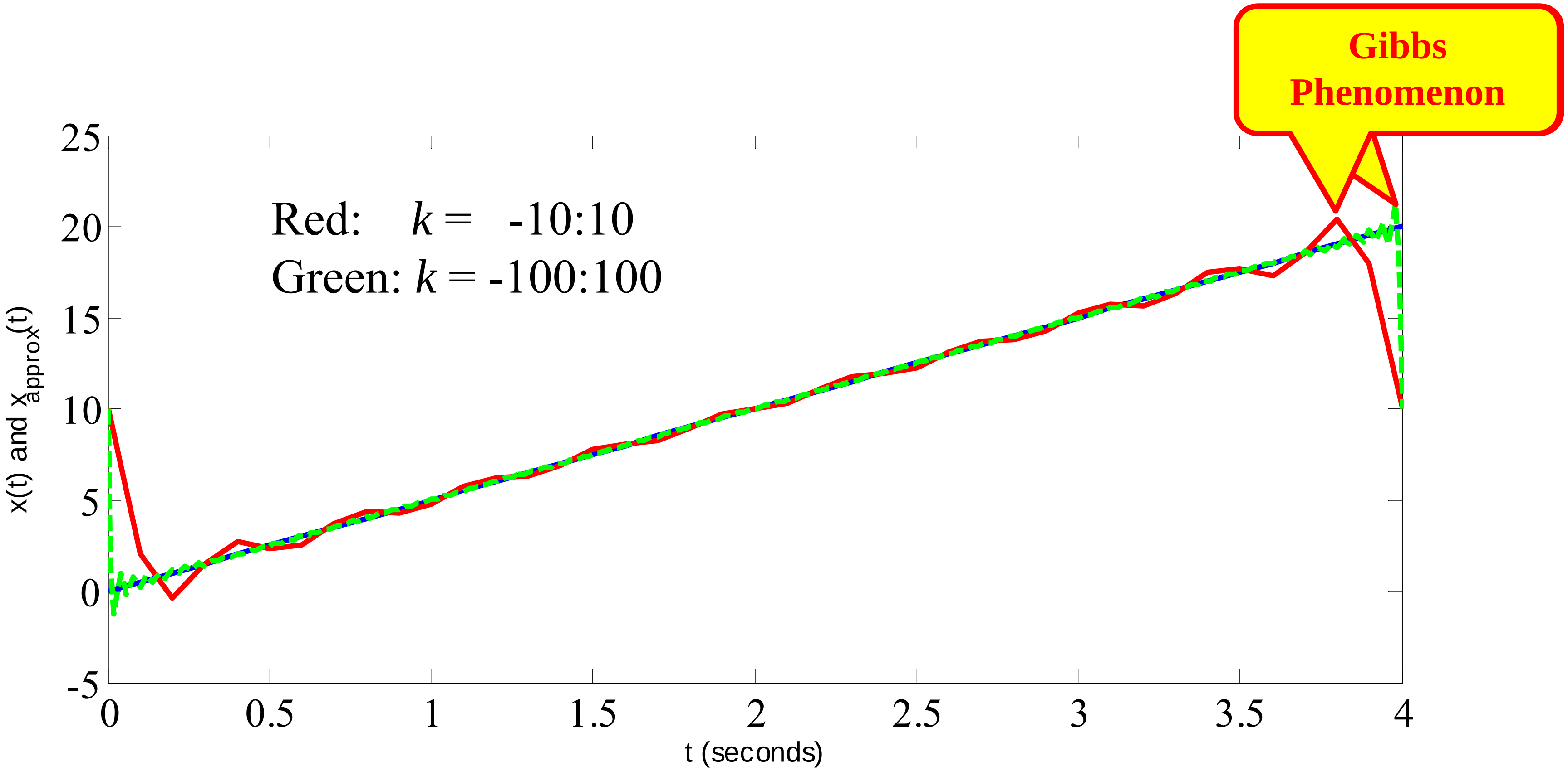
$$x(t) \approx \sum_{k=-K}^{K} c_k e^{jk2\pi f_0 t}$$

```
%  Assume we have these quantities found previously:   c, wo, K, Ts
t = 0:Ts:T;   % computes over one period… but could compute over larger range
x_apprx = zeros(size(t));   % sets up vector of zeros as first "partial sum"
for k = (-K):K     % loop through "all" coefficients
        x_apprx = x_apprx + c(k+K+1)*exp(j*k*wo*t);    % Add current term to partial sum
end
x_apprx = real(x_apprx);    % theory says imaginary parts cancel… so enforce this in case
                                                            % of numerical round-off issues
```

Must be appropriate for largest frequency term used



Improving the Approximate Signal by using More Terms



- **Note that more terms gives a better approximation… but there is still "ringing" error at the discontinuities regardless of how many terms are included.**
- **This is called the Gibbs Phenomenon.**