So far…

- Signals to sequences
- Convolution $y(n)$ between sequences $a(n)$ and $b(n)$ is defined as

$$y(n) = \sum_{k=-\infty}^{\infty} a(k)\ b(n-k)$$

or

$$y(n) = a(n) \star b(n)$$

- Correlation $r_{x,y}(l)$ between two sequences $x(n)$ and $y(n)$

$$r_{x,y}(l) = x(l) \star y(-l)$$

- Impulse response of a system $h(n)$
  - Where $h(n)$ is the *unit sample* or *impulse* response of the LTI system
  - System is stable if

$$\sum_{-\infty}^{+\infty} |h(n)| < \infty$$

  - In general, the response sequence $y(n)$ to the input sequence $x(n)$ can be re-written as

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)\ h(n-k) = x(n) \star h(n)$$

So far…

- Differential to difference equations

$$y(t) = \frac{dx(t)}{dt} \quad \rightarrow \quad y(n) = x(n) - x(n-1)$$

  – General form of a difference equation

$$\sum_{k=0}^{N} a_k \, y(n-k) = \sum_{m=0}^{M} b_m x(n-m)$$

  – MATLAB's `filter` command to numerically solve difference equations:

```
>> y = filter(b, a, x)
```

  – In particular the impulse response $h(n)$ of a system can be found

```
>> h = filter(b, a, delta)
```

Digital Signal Processing 2

- In the analog domain, the Laplace transform $\mathcal{L}$
  - Relates time-functions to frequency-dependent functions
- For the digital domain, the $\mathcal{Z}$ transform
  - Relates time-sequences to (a different, but related type of) frequency-dependent function

The $\mathcal{Z}$ Transform

- This discrete-time equivalent of the *Laplace* transform is defined as

$$X(z) = \mathcal{Z}[x(n)] = \sum_{n=-\infty}^{\infty} x(n)\, z^{-n}$$

$$x(n) = \mathcal{Z}^{-1}[X(z)] = \frac{1}{2\pi j} \oint X(z)\; z^{n-1} dz$$

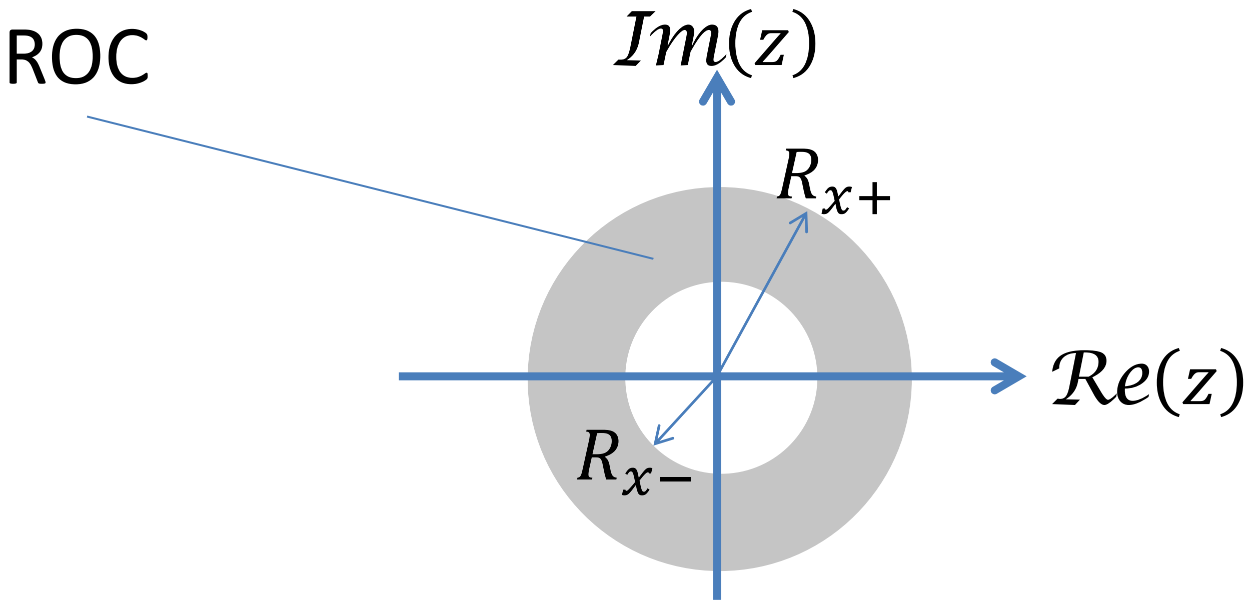
  where $z = |z|e^{j\omega}$ is the complex frequency.

- The values of z for which the sum converges define a region in the z-plane referred to as the *region of convergence* (ROC).

Region of convergence (ROC)

The set of z values for which $X(z)$ exists is called the region of convergence (ROC)

$$R_{x-} < |z| < R_{x+}$$



The $\mathcal{Z}$ Transform

This transformation is useful in

1. Solving constant coefficient difference equations
2. Evaluating the response of an LTI system to a given input, and
3. Designing linear filters

Example

Geometric series:

$$\sum_{n=0}^{\infty} \propto^n = \frac{1}{1-\propto} \text{ if } |\propto| < 1$$

Let $x(n) = 1$ for $n = 0,1,2 \ldots$.
Find its $\mathcal{Z}$ transform and its ROC.

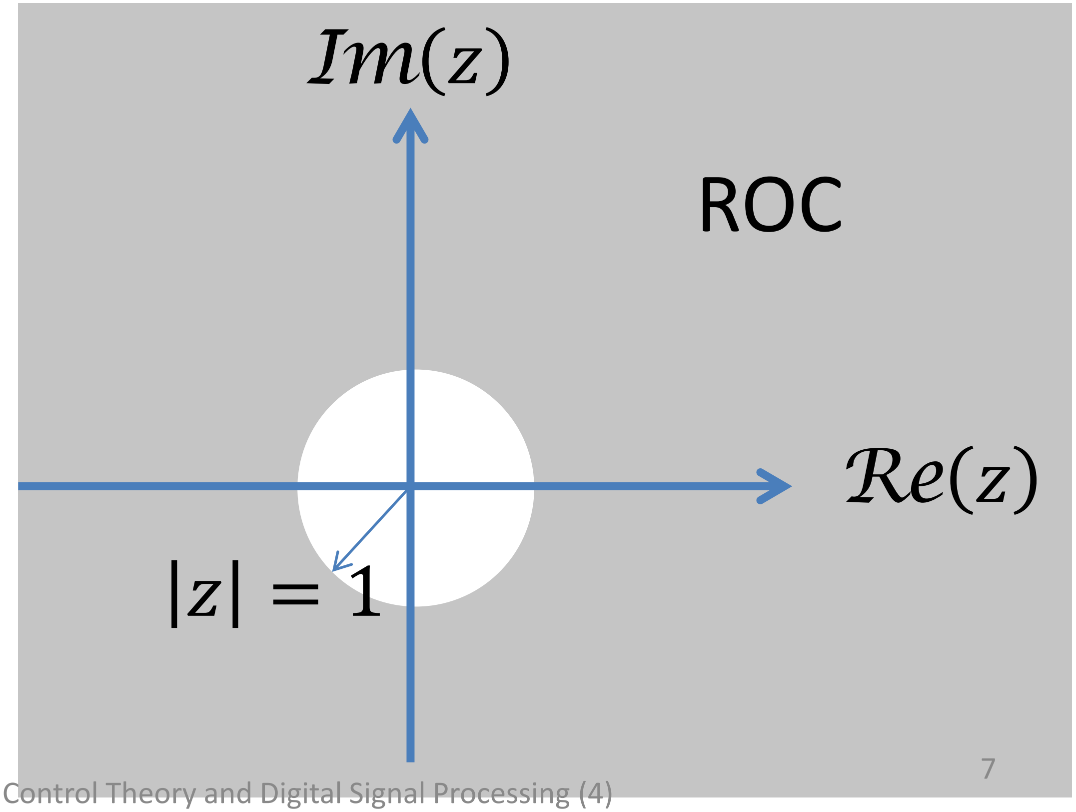
$$X(z) = \sum_{n=0}^{\infty} x(n)\, z^{-n} = \sum_{n=0}^{\infty} z^{-n} = 1 + z^{-1} + z^{-2} + \cdots$$

$$= \frac{1}{1 - z^{-1}}$$

if $|z^{-1}| < 1$ or $|z| > 1$

Transfer function:
zero at the origin,
pole at 1

$$X(z) = \frac{z}{z-1}$$



Example

Let $x(n) = 2^n$ for $n = 0,1,2 \ldots$ Find its $\mathcal{Z}$ transform and its ROC.

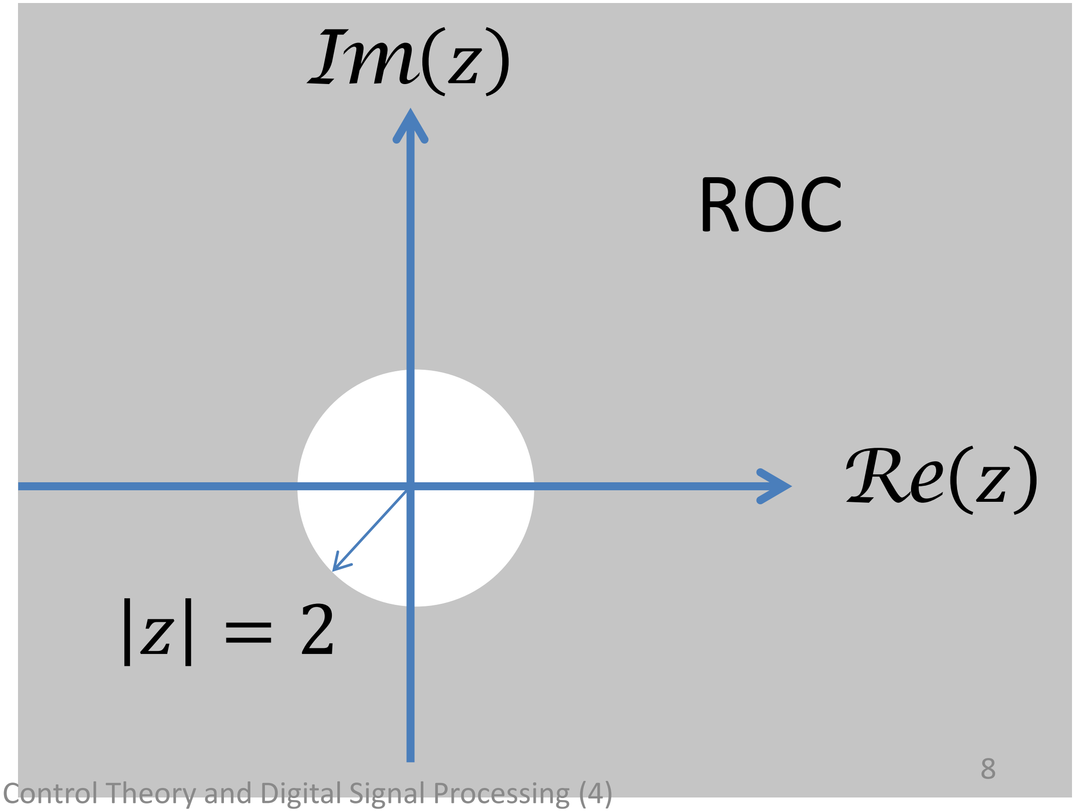
$$X(z) = \sum_{n=0}^{\infty} x(n)\, z^{-n} = \sum_{n=0}^{\infty} 2^n\, z^{-n} = 1 + 2\, z^{-1} + 4\, z^{-2} + \cdots$$

$$= \frac{1}{1 - 2\, z^{-1}}$$

if $|2\, z^{-1}| < 1$ or $|z| > 2$

Transfer function: zero at the origin, pole at 2

$$X(z) = \frac{z}{z - 2}$$



Example

Let $x_1(n) = a^n\, u(n), 0 < |a| < \infty$. Find its Z transform and its ROC.

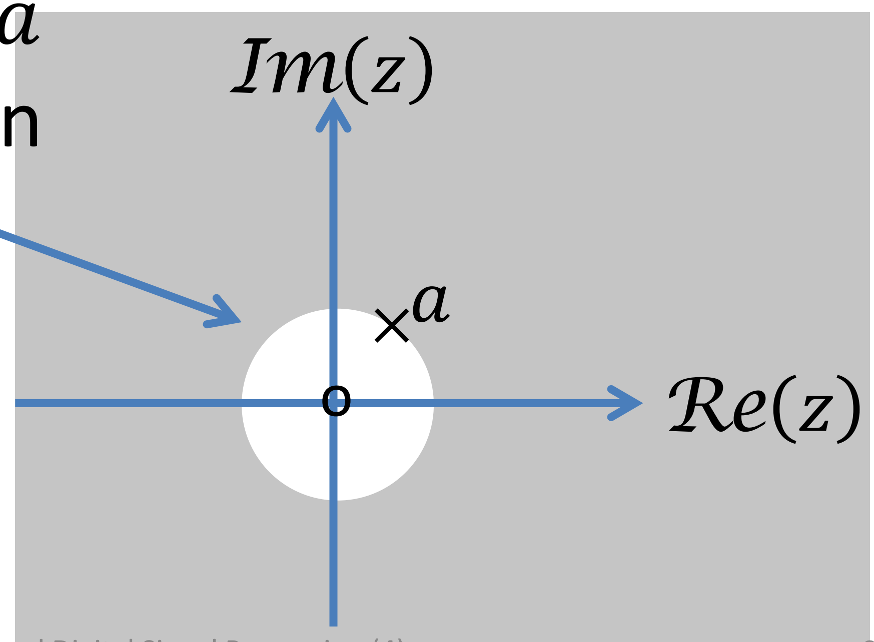
$$X_1(z) = \sum_{0}^{\infty} a^n\, z^{-n} = \sum_{0}^{\infty} \left(\frac{a}{z}\right)^n = \frac{1}{1 - az^{-1}}$$

if $\left|\frac{a}{z}\right| < 1$

Pole at $z = a$

Zero at origin

$$X_1(z) = \frac{z}{z - a}$$



In general

Many of the signals in DSP have $\mathcal{Z}$ transforms that are rational (ratio of two polynomials) functions of z^{-1}:

$$X(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{a_0 + a_1 z^{-1} + \cdots + a_N z^{-N}}$$
$$= C\,\frac{\prod_{l=1}^{N}(z - z_l)}{\prod_{k=1}^{N}(z - p_k)}$$

where p_k is the k-th pole and z_l is the l-th zero of $X(z)$. Each pole is indicated by an ”x” and each zero by an “o” in the z-plane.

Special properties of the $\mathcal{Z}$ Transform

- Convolution
  – Given two sequences $x_1(n)$ and $x_2(n)$, their time-domain *convolution* becomes a *multiplication* process in the frequency domain

$$\mathcal{Z}[x_1(n) * x_2(n)] = X_1(z) \cdot X_2(z)$$

- Sample shifting

$$\mathcal{Z}[x(n-n_0)] = z^{-n_0} X(z)$$

(Some) $\mathcal{Z}$ Transform pairs

$x(n)$	$X(z)$	ROC
$\delta(n)$	1	*Any z*
$u(n)$	$\frac{1}{1-z^{-1}}$	$\|z\| > 1$
$a^n\,u(n)$	$\frac{1}{1-a\,z^{-1}}$	$\|z\| > \|a\|$
$n\,a^n\,u(n)$	$\frac{a\,z^{-1}}{(1-a\,z^{-1})^2}$	$\|z\| > \|a\|$

And back to sequences: The Inverse $\mathcal{Z}$ Transform

- Just like in the Laplace domain
- Use the partial fraction method to reduce a complex $X(z)$ to simpler parts.
- Use the table of transform pairs to determine the sequence.
- In general

If $M \geq N$

$$X(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{a_0 + a_1 z^{-1} + \cdots + a_N z^{-N}}$$

$$= \sum_{k=1}^{N} \frac{R_k}{1 - p_k z^{-1}} + \sum_{k=0}^{M-N} C_k \, z^{-k}$$

Example

Compute the inverse $\mathcal{Z}$-transform of

$$X(z) = \frac{z}{3\,z^2 - 4\,z + 1}, |z| > 1$$

Sol:

Re-write $X(z)$ in terms of powers of z^{-1}.

$$X(z) = \frac{z^{-1}}{3 - 4\,z^{-1} + z^{-2}}$$

At the MATLAB prompt

```
>> b=[0 1];a=[3 -4 1];
>> [R,p,C]=residuez(b,a)
R =
     0.5000
    -0.5000
p =
     1.0000
     0.3333
C =
       []
```

$$X(z) = \frac{1/2}{1 - z^{-1}} - \frac{1/2}{1 - \frac{1}{3}z^{-1}}$$

Using the table of transform pairs

$$x(n) = \frac{1}{2}\,u(n) - \frac{1}{2}\left(\frac{1}{3}\right)^n u(n)$$

Example

Compute the inverse $\mathcal{Z}$-transform of

$$X(z) = \frac{1}{(1-0.9\,z^{-1})^2(1+0.9\,z^{-1})}, |z| > 0.9$$

Sol:

Using MATLAB to do the partial fraction

```
>> b = 1; a = poly([0.9, 0.9, -0.9])

a =

    1.0000   -0.9000   -0.8100    0.7290
```

$$X(z) = \frac{1}{1+0.9\,z^{-1}-0.81\,z^{-2}+0.729\,z^{-3}}$$

Example

```
>> [R,p,C]=residuez(b,a)
R =
   0.2500
   0.2500 + 0.0000i
   0.5000 - 0.0000i
p =
  -0.9000
   0.9000 + 0.0000i
   0.9000 - 0.0000i
C =
     []
```

Example

Sample shifting

$$X(z) = \frac{0.25}{1-0.9\,z^{-1}} + \frac{0.5}{(1-0.9\,z^{-1})^2} + \frac{0.25}{1+0.9\,z^{-1}}$$

$$= \frac{0.25}{1-0.9\,z^{-1}} + \frac{0.5}{0.9}\,z\,\frac{(0.9\,z^{-1})}{(1-0.9\,z^{-1})^2} + \frac{0.25}{1+0.9\,z^{-1}}$$

Using the $\mathcal{Z}$ Transform pair table

$$x(n) = 0.25 \cdot 0.9^n \cdot u(n) + \frac{0.5}{0.9} \cdot (n+1) \cdot 0.9^{(n+1)} \cdot u(n+1) + 0.25 \cdot (-0.9)^n \cdot u(n)$$

Exercise

Determine the $\mathcal{Z}$-transform of the impulse response

$$h(n) = 2\,\delta(n-2) + 3\,u(n-3)$$

Exercise

Determine the $\mathcal{Z}$-transform of the impulse response

$$h(n) = 2\,\delta(n-2) + 3\,u(n-3)$$

Sol:

$$\mathcal{Z}[h(n)] = 2\mathcal{Z}[\delta(n-2)] + 3\mathcal{Z}[u(n-3)]$$

Using the *sample shift* property

$$\mathcal{Z}[h(n)] = 2\,z^{-2}\mathcal{Z}[\delta(n)] + 3\,z^{-3}\mathcal{Z}[u(n)]$$

$$H(z) = 2\,z^{-2} + 3\,z^{-3}\frac{1}{1-z^{-1}}$$

$$H(z) = \frac{2\,z^{-2} - 2\,z^{-3} + 3z^{-3}}{1-z^{-1}}$$

System function $H(z)$

The system function $H(z)$ is simply the $\mathcal{Z}$ transform of the impulse response of the system

$$H(z) = \mathcal{Z}[h(n)] = \sum_{-\infty}^{+\infty} h(n) z^{-n}$$

This means that, given input $\mathrm{X}(z)$ the output $\mathrm{Y}(z)$ is

$$Y(z) = H(z)X(z)$$

System function $H(z)$ from a difference equation

When an LTI system is represented by the difference equation

$$y(n) + \sum_{k=1}^{N} a_k \, y(n-k) = \sum_{l=0}^{M} b_l \, x(n-l)$$

it can be shown that

$$H(z) = \frac{\sum_{l=0}^{M} b_l \; z^{-l}}{1 + \sum_{k=1}^{N} a_k \; z^{-k}}$$

where $a_0 = 1$

System function $H(z)$ and MATLAB implementation

The MATLAB implementation, given

$$H(z) = \frac{\sum_{l=0}^{M} b_l \; z^{-l}}{1 + \sum_{k=1}^{N} a_k \; z^{-k}}$$

The impulse response $h(n)$ is simply

```
>> h = filter(b, a, delta)
```

While the response $y(n)$ to input $x(n)$ is

```
>> y = filter(b, a, x)
```

For example

Given the LTI system represented by the difference equation, the determine the impulse response $h(n)$.

$$y(n) = 0.9\, y(n-1) + x(n)$$

Sol: let's find the system function $H(z)$ first.

$$y(n) - 0.9\, y(n-1) = x(n)$$

Taking the $\mathcal{Z}$ transform

$$\mathcal{Z}[y(n)] - 0.9\mathcal{Z}[y(n-1)] = \mathcal{Z}[x(n)]$$

$$Y(z) - 0.9\, z^{-1}\, Y(z) = X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - 0.9\, z^{-1}}$$

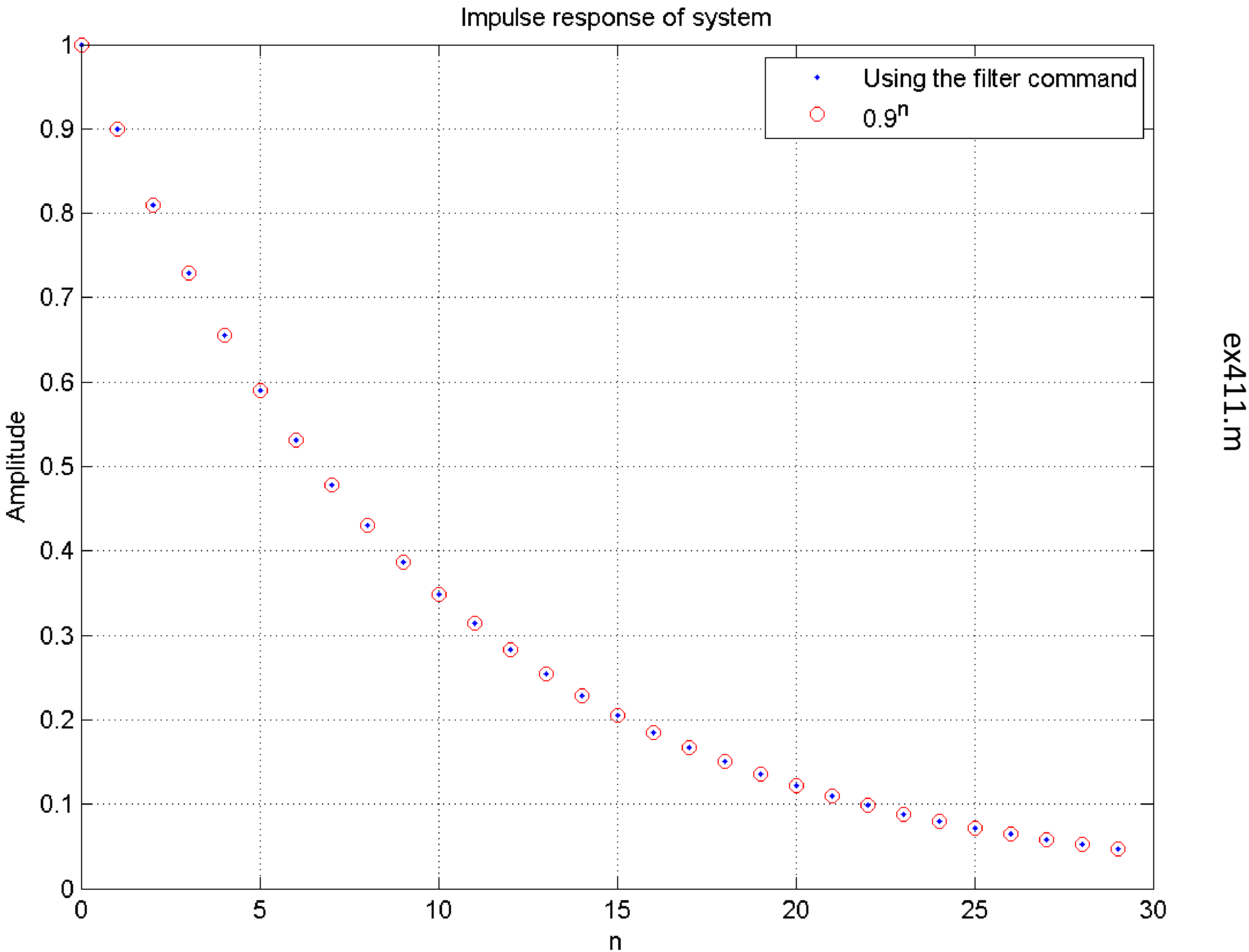
Taking the inverse transform

$$h(n) = \mathcal{Z}^{-1}[H(z)] = 0.9^{n}\, u(n)$$

Let's verify with MATLAB

```
>> b = [1]; a = [1 -0.9];
>> h = filter(b,a,delta);
```

and lets plot the two responses.



Impulse response of system
Using the filter command
0.9n
Amplitude
n
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25
30
ex411.m

To recap

The $\mathcal{Z}$ transform is the digital equivalent of the Laplace transform:

1. It facilitates the solving of constant coefficient difference equations
2. It allows to easily evaluate the system's response
3. It is critical in designing linear filters

MATLAB commands used

– `filter`, `residuez`

A different transform: DTFT $\mathcal{F}$

- A very different but very useful representation of a sequence or system is the <u>Discrete-time Fourier Transform (DTFT)</u>
- Setting $|z| = 1$ in the $\mathcal{Z}$-transform

$$X(z) = \mathcal{Z}[x(n)] = \sum_{n=-\infty}^{\infty} x(n)\, z^{-n}$$

$$x(n) = \mathcal{Z}^{-1}[X(z)] = \frac{1}{2\pi j} \oint X(z)\;\, z^{n-1} dz$$

where $z = |z|e^{j\omega}$ is the complex frequency.

A different transform: DTFT $\mathcal{F}$

- The DTFT of sequence $x(n)$ is defined as

$$X(e^{j\omega}) = \mathcal{F}[x(n)] = \sum_{n=-\infty}^{\infty} x(n)\, e^{-j\omega n}$$

$$x(n) = \mathcal{F}^{-1}[X(e^{j\omega})] = \frac{1}{2\pi}\int_{-\pi}^{+\pi} X(e^{j\omega})\, e^{j\omega n} d\omega$$

where

- $X(e^{j\omega})$ is a complex valued function
- ω is a digital frequency ranging from $-\pi$ to $+\pi$

Special property of the DTFT

Just like the $\mathcal{Z}$ transform

Convolution

- Given two sequences $x_1(n)$ and $x_2(n)$, their convolution is a *multiplication* process in the frequency domain

$$\mathcal{F}[x_1(n) * x_2(n)] = \mathcal{F}[x_1(n)] \cdot \mathcal{F}[x_2(n)]$$
$$= X_1(e^{j\omega}) \cdot X_2(e^{j\omega})$$

Frequency domain representation of LTI systems

The DTFT of the unit sample response is called the Frequency Response or Transfer Function of an LTI system

$$H\left(e^{j\omega}\right) = \sum_{n=-\infty}^{\infty} h(n)\, e^{-j\omega n}$$

Frequency response from difference equations

When an LTI system is represented by the difference equation

$$y(n) + \sum_{l=1}^{N} a_l \, y(n-l) = \sum_{m=0}^{M} b_m \, x(n-m)$$

Then

$$H\left(e^{j\omega}\right) = \frac{\sum_{m=0}^{M} b_m e^{-j\omega m}}{1 + \sum_{l=1}^{N} a_l e^{-j\omega l}}$$

Example

Given difference equation

$$y(n) = 0.8\, y(n-1) + x(n)$$

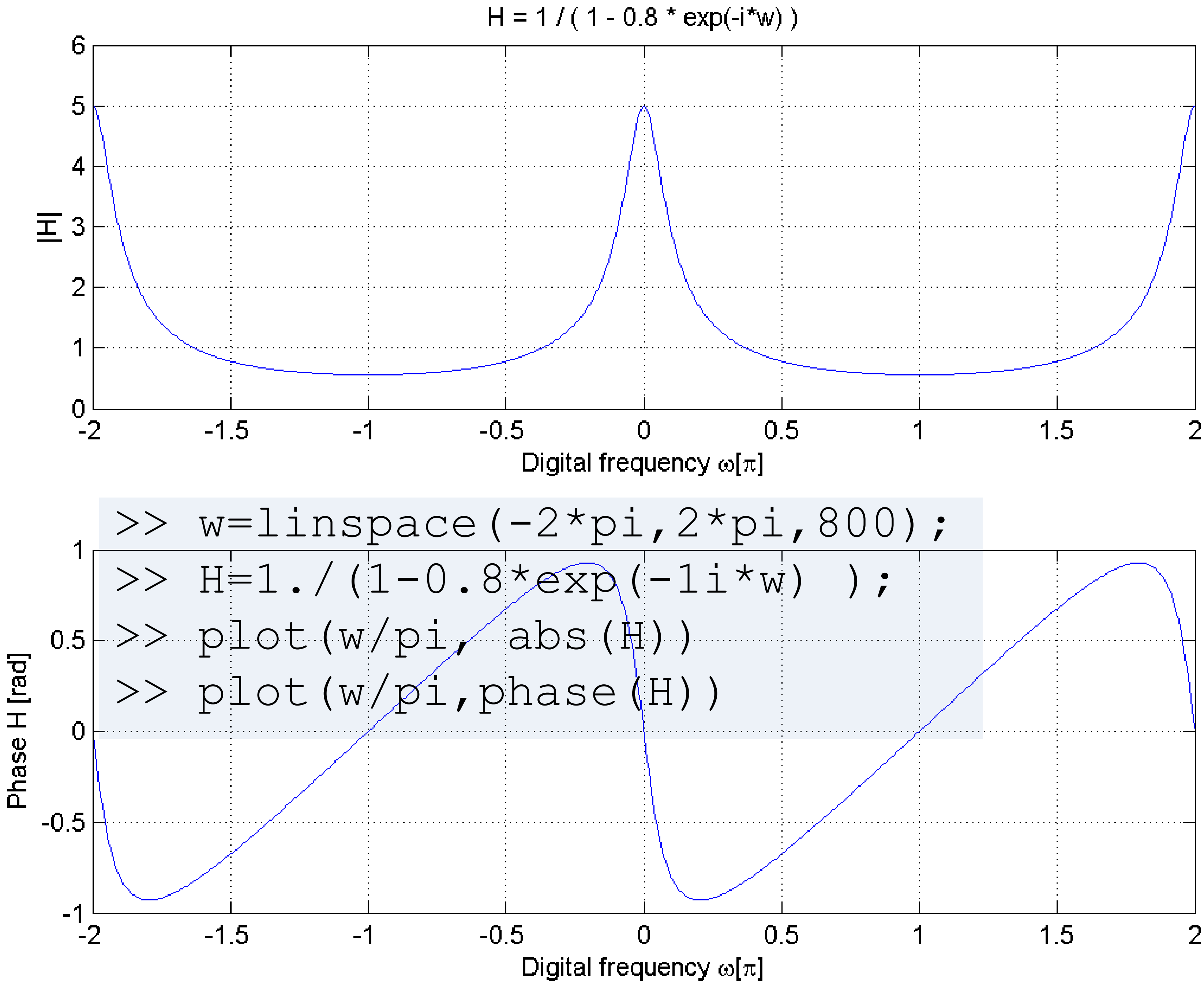
determine transfer function $H\left(e^{j\omega}\right)$ and plot its magnitude and phase

Example

$$y(n) = 0.8\,y(n-1) + x(n)$$

$$a_0 = 1, a_1 = -0.8, b_0 = 1$$

$$H\left(e^{j\omega}\right) = \frac{1}{1 - 0.8e^{-j\omega}}$$




```
>> w=linspace(-2*pi,2*pi,800);
>> H=1./(1-0.8*exp(-1i*w) );
>> plot(w/pi, abs(H))
>> plot(w/pi,phase(H))
```

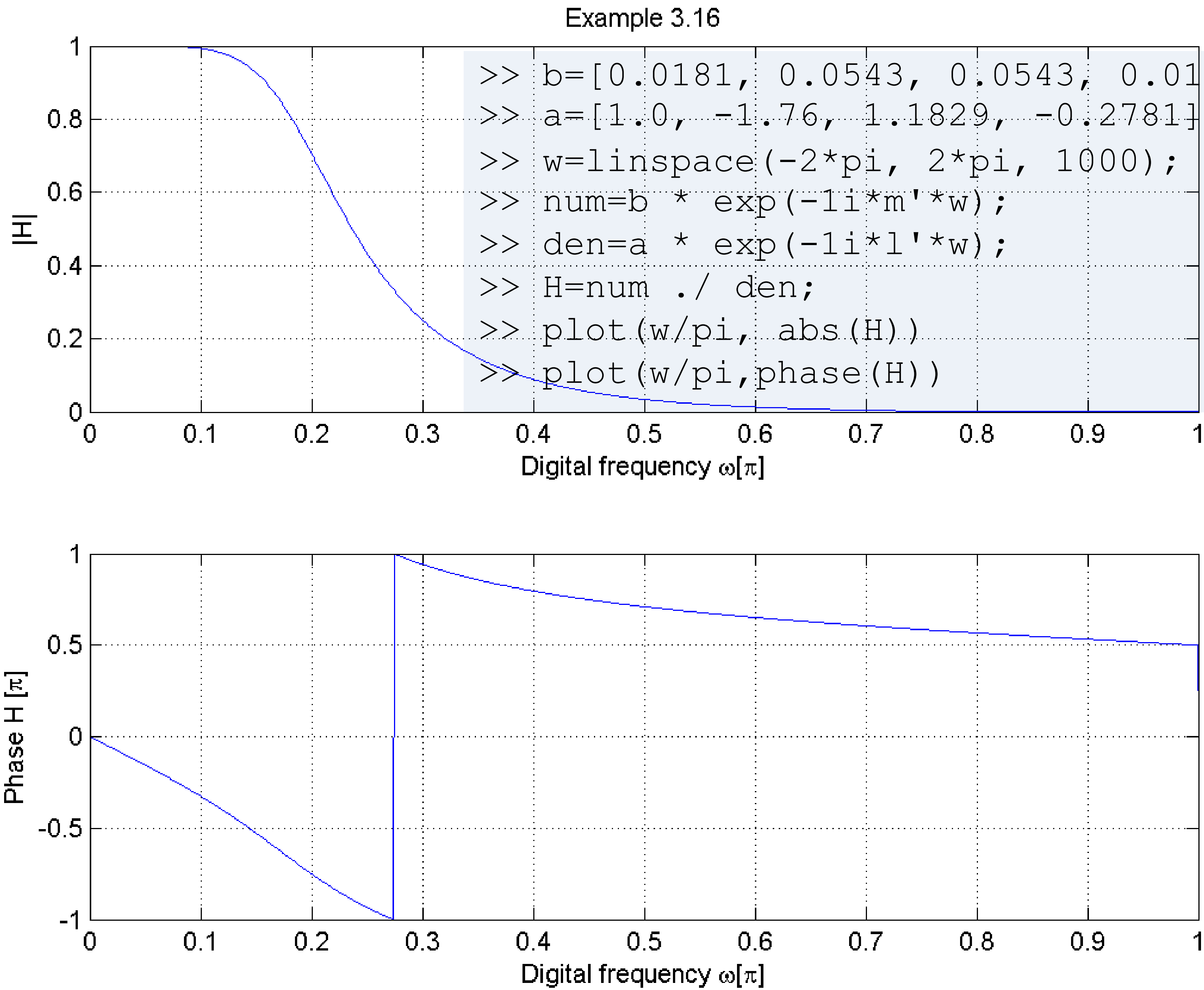
dtft_example3.m

Example 3.16

A 3rd order low pass filter is described by the difference equation

$$y(n) = 0.0181\,x(n) + 0.0543\,x(n-1) + 0.0543\,x(n-2) + 0.0181\,x(n-3) + 1.76\,y(n-1) - 1.1829\,y(n-2) + 0.2781\,y(n-3)$$

Plot the magnitude and the phase response of this filter and verify that it is a low pass filter.




```
>> b=[0.0181, 0.0543, 0.0543, 0.0181];
>> a=[1.0, -1.76, 1.1829, -0.2781];
>> w=linspace(-2*pi, 2*pi, 1000);
>> num=b * exp(-1i*m'*w);
>> den=a * exp(-1i*l'*w);
>> H=num ./ den;
>> plot(w/pi, abs(H))
>> plot(w/pi,phase(H))
```

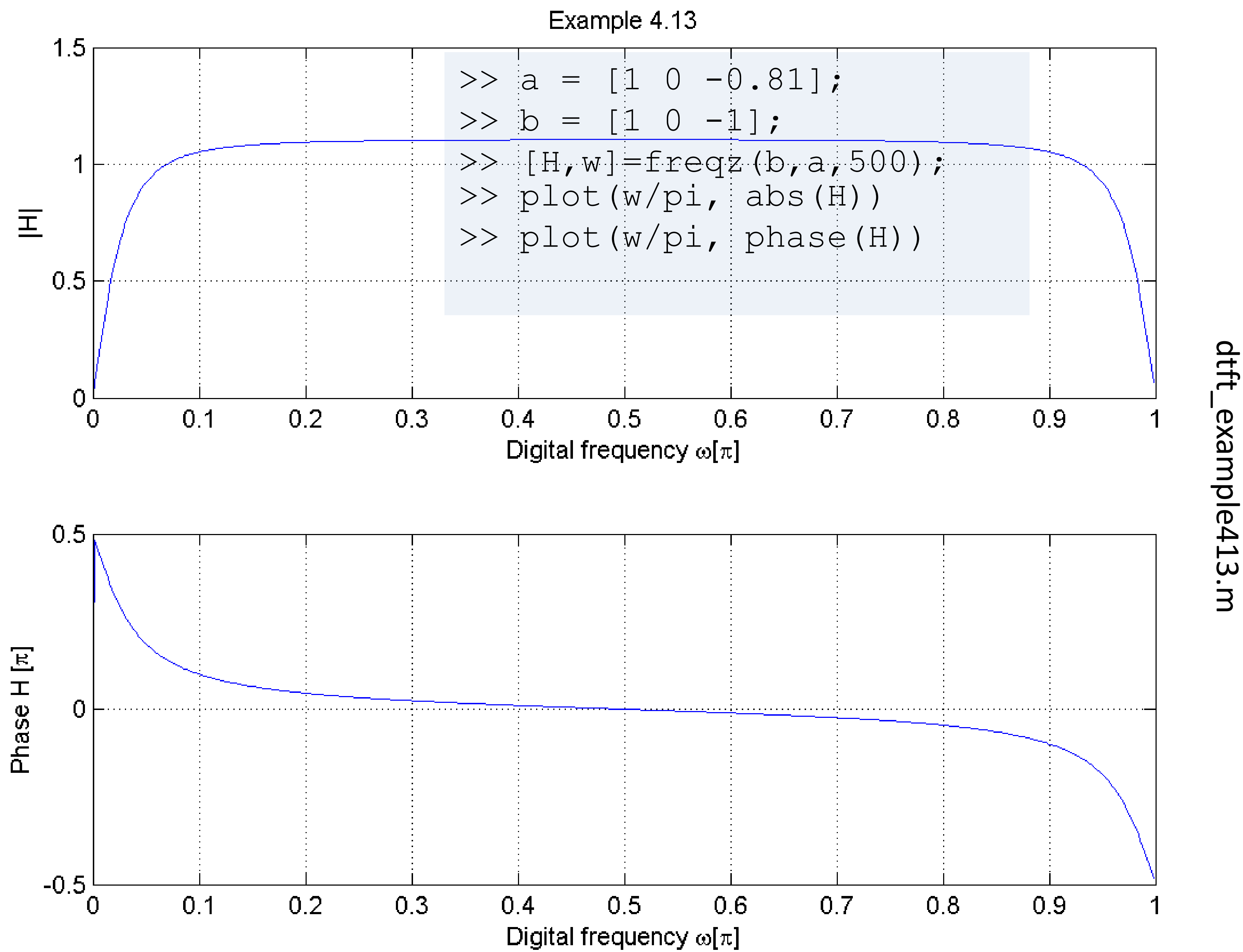
dtft_example316.m

Example 4.13

An LTI system is described by the following difference equation

$$y(n) = 0.81\, y(n-2) + x(n) - x(n-2)$$

Plot the magnitude and the phase response of this filter.



Example 4.13
1.5
1
0.5
0
|H|
>> a = [1 0 -0.81];
>> b = [1 0 -1];
>> [H,w]=freqz(b,a,500);
>> plot(w/pi, abs(H))
>> plot(w/pi, phase(H))
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Digital frequency ω[π]
0.5
0
-0.5
Phase H [π]
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Digital frequency ω[π]
dtft_example413.m

Another transform: The Discrete Fourier Transform (DFT)

- The FFT falls into this category
- Why more transforms? What is the problem?
  - The DTFT and the $\mathcal{Z}$ transform are not numerically computable transforms
    - They have infinite sums at uncountably infinite frequencies
- The Discrete Fourier Transform (DFT)
  - Obtained by sampling the Discrete-Time Fourier Transform (DTFT) in the frequency domain
  - "the DFT is just equally-spaced samples of the DTFT"
  - Time-consuming numerical computation
- <u>Fast-Fourier Transform</u>
  - Algorithm for the efficient computation of DFTs

Example: highlighting the difference between the DTFT and DFT

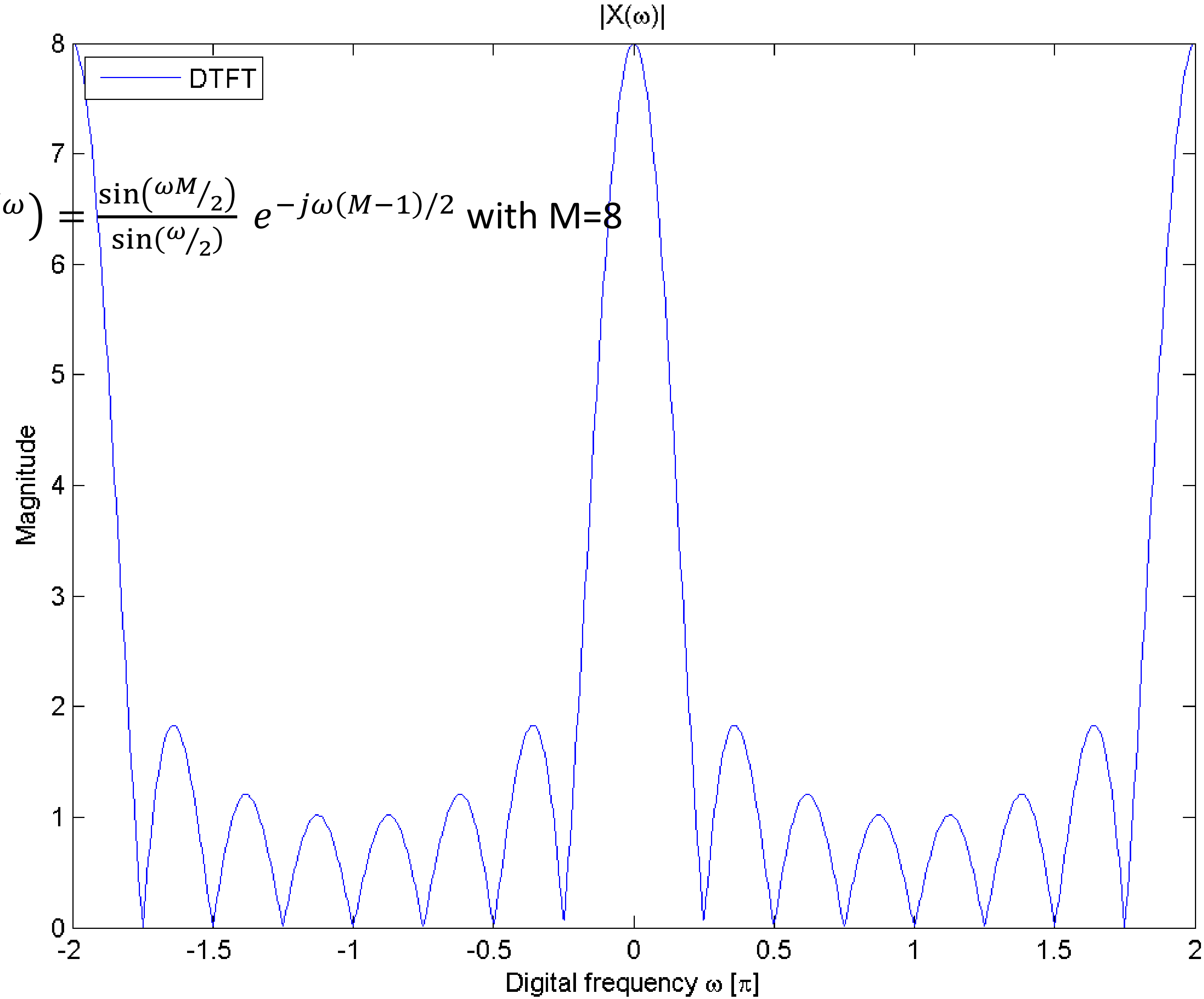
Let

$$x(n) = 1 \text{ for } 0 \leq n \leq 8$$

The corresponding DTFT is

$$X\left(e^{j\omega}\right) = \frac{\sin\left(\omega M/_{2}\right)}{\sin\left(\omega/_{2}\right)}\, e^{-j\omega(M-1)/2}$$

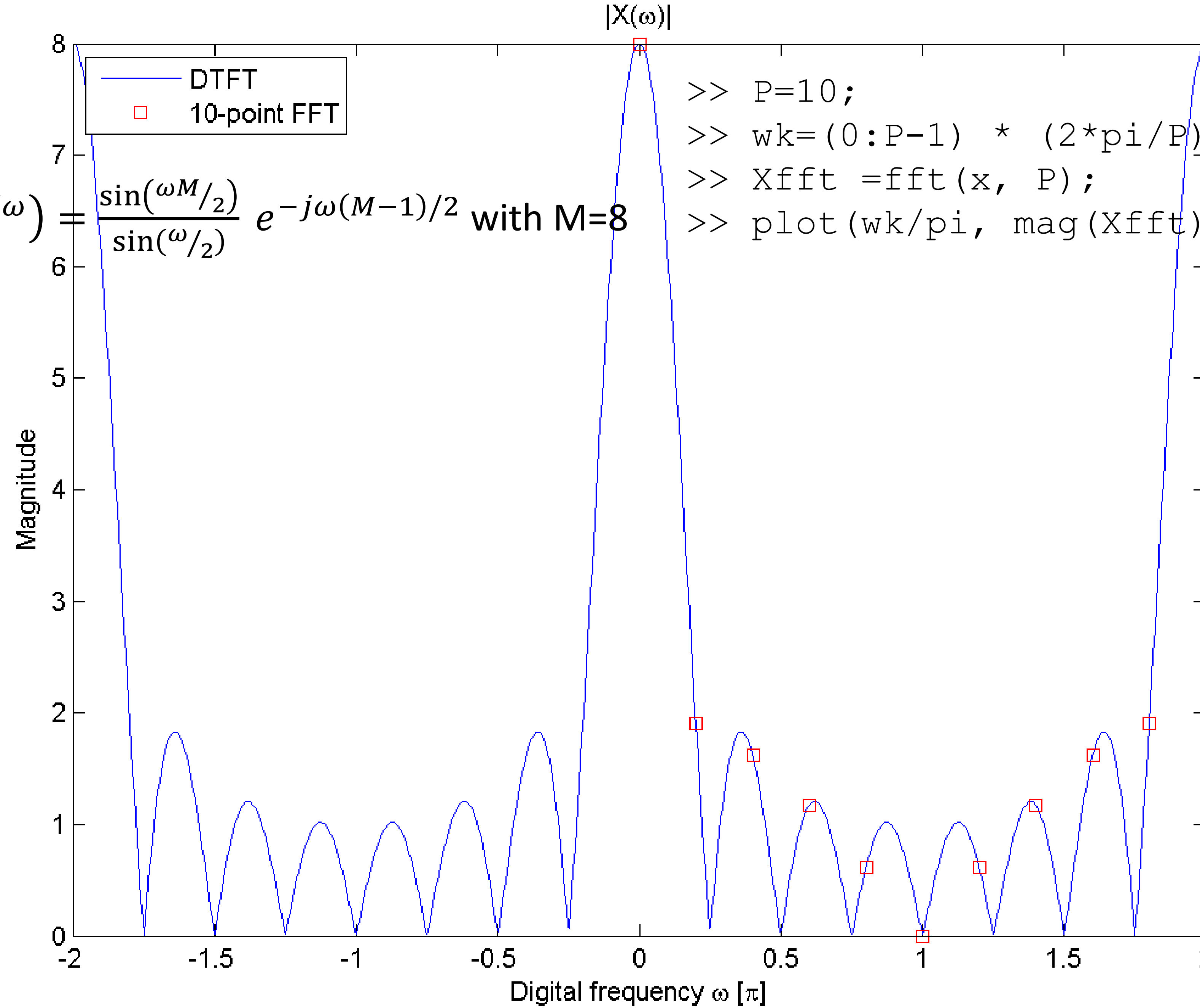
$$X\left(e^{j\omega}\right) = \frac{\sin\left(\omega M/_2\right)}{\sin\left(\omega/_2\right)}\, e^{-j\omega(M-1)/2}$$ with M=8



dtft_example.m

$$X\left(e^{j\omega}\right) = \frac{\sin\left(\omega M/_2\right)}{\sin\left(\omega/_2\right)}\ e^{-j\omega(M-1)/2} \text{ with M=8}$$

```
>> P=10;
>> wk=(0:P-1) * (2*pi/P);
>> Xfft =fft(x, P);
>> plot(wk/pi, mag(Xfft),'rs')
```

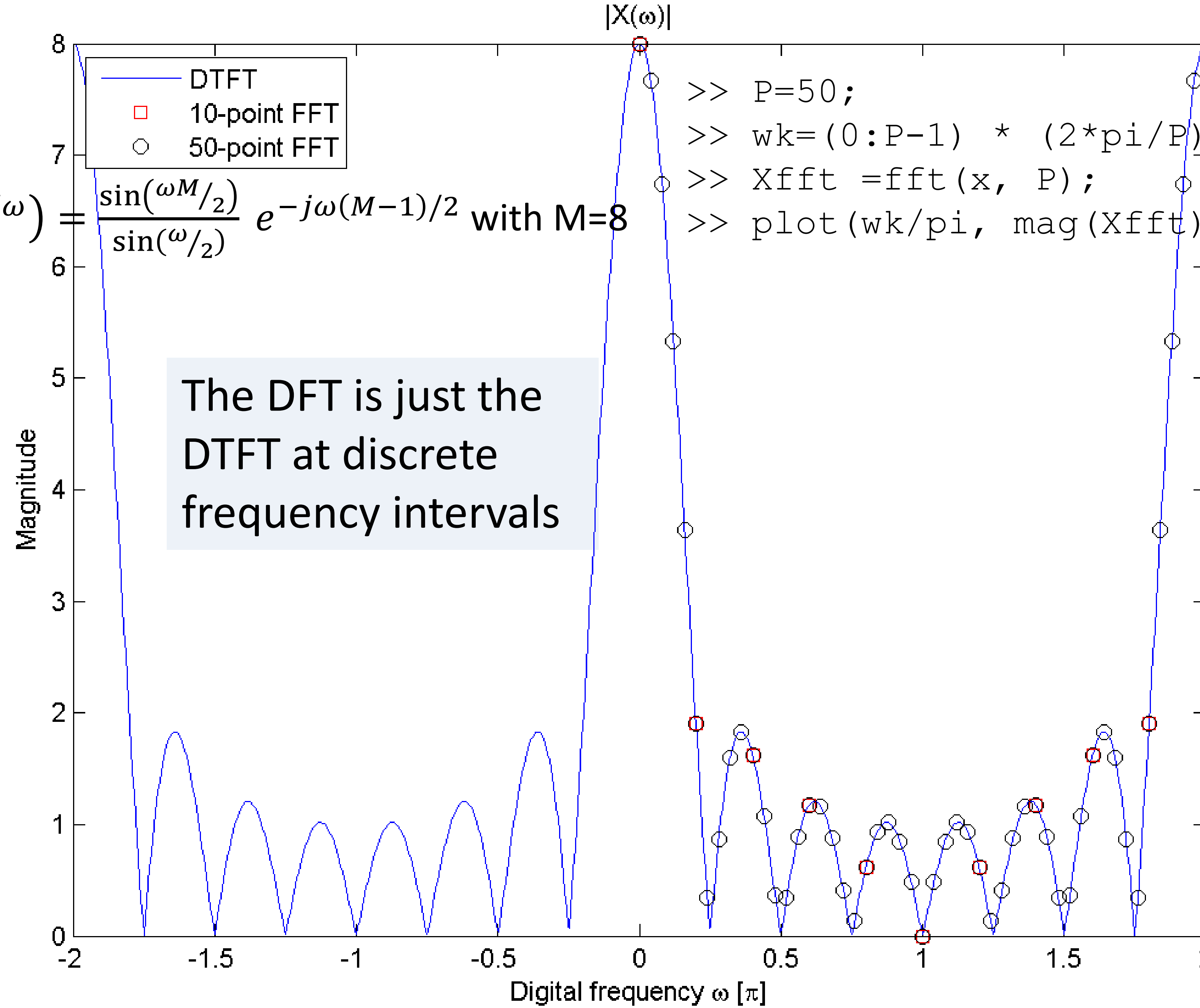


dtft_example.m

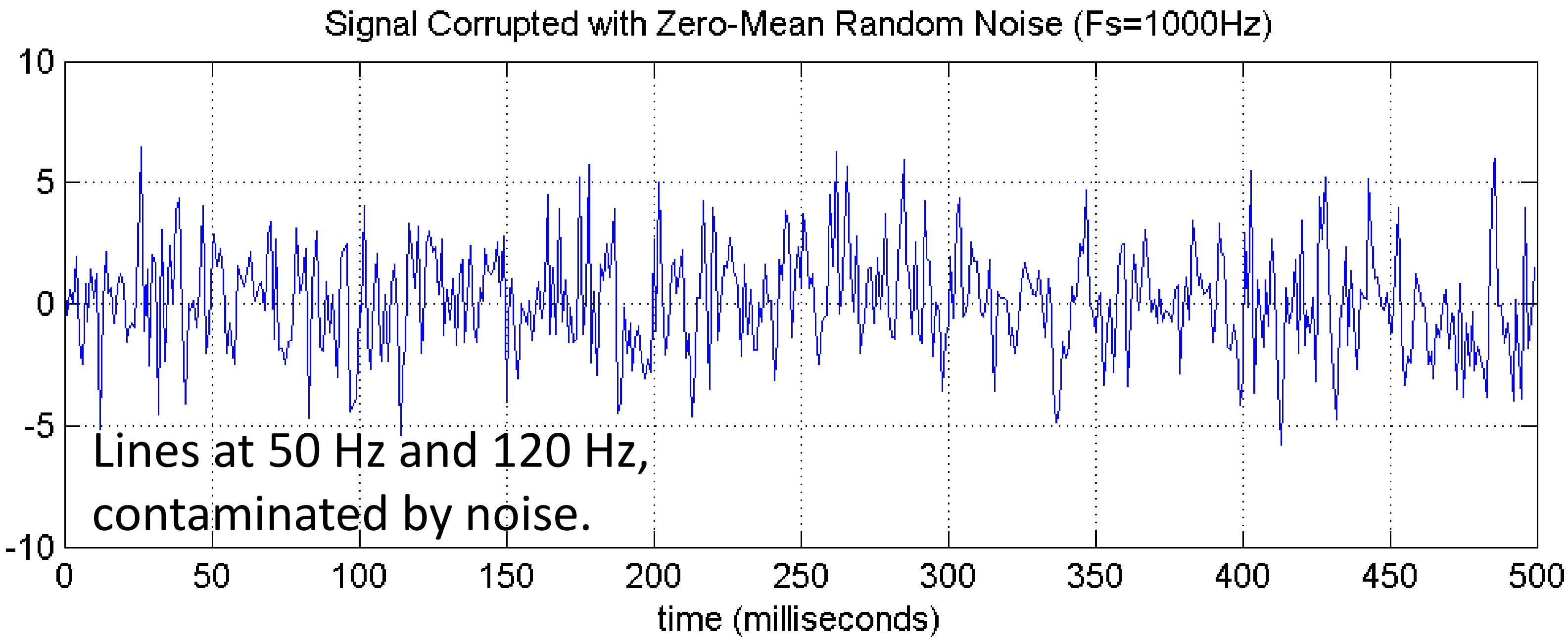
$$X\left(e^{j\omega}\right) = \frac{\sin\left(\omega M/_2\right)}{\sin\left(\omega/_2\right)}\, e^{-j\omega(M-1)/2} \text{ with M=8}$$

```
>> P=50;
>> wk=(0:P-1) * (2*pi/P);
>> Xfft =fft(x, P);
>> plot(wk/pi, mag(Xfft),'rs')
```

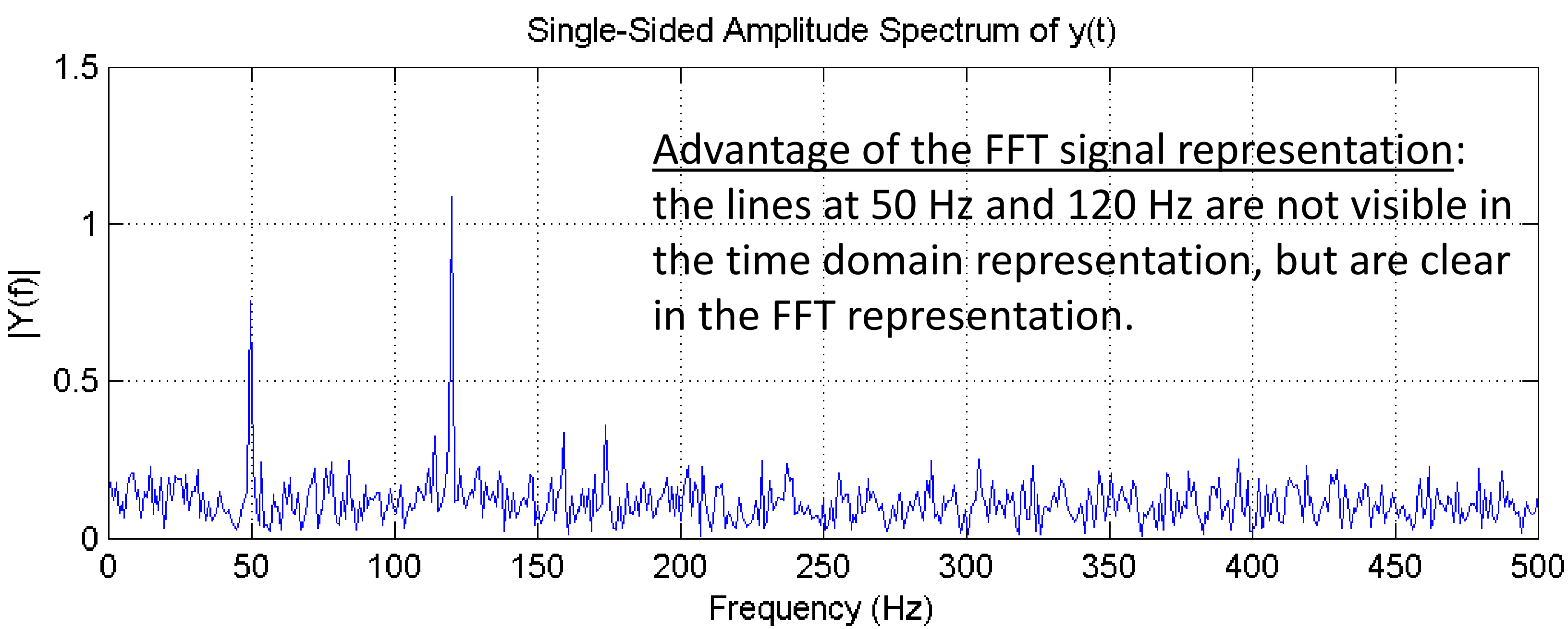
The DFT is just the DTFT at discrete frequency intervals



dtft_example.m



Lines at 50 Hz and 120 Hz, contaminated by noise.

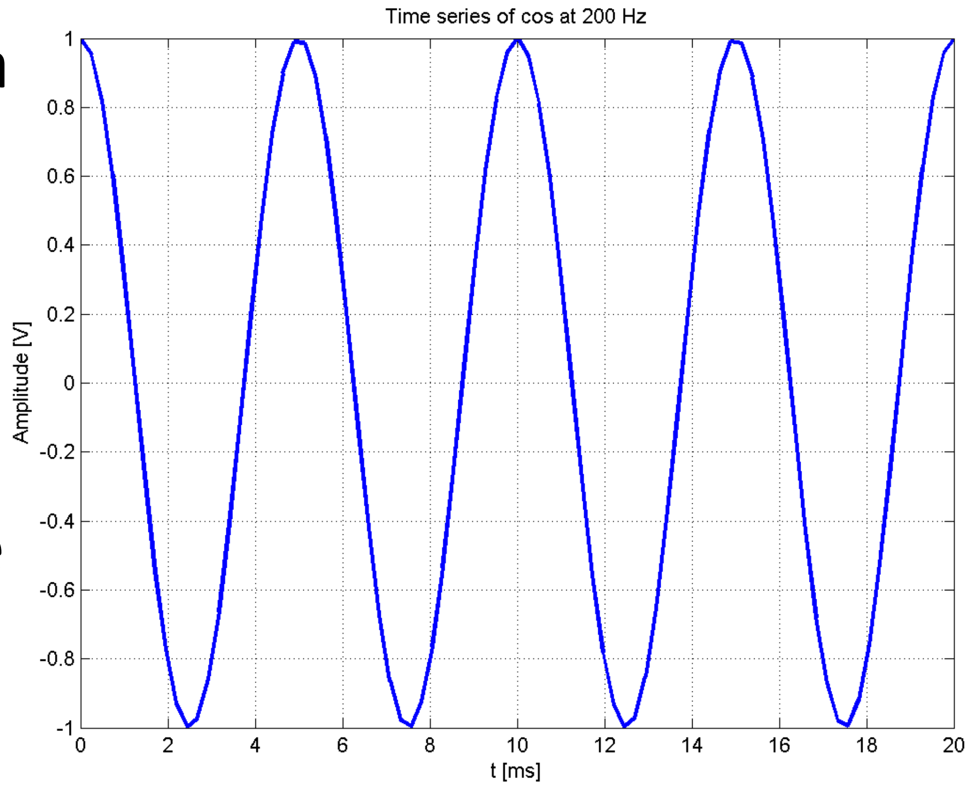


<u>Advantage of the FFT signal representation</u>: the lines at 50 Hz and 120 Hz are not visible in the time domain representation, but are clear in the FFT representation.

dtft_example.m

Power Spectral Density

- A graphical representation to easily determine the power of a signal over a particular frequency band.
- Uses the `fft` algorithm
- Unfortunately there are many conventions for the normalization, can be confusing…
- Let's use the example of a cosine at 200 Hz



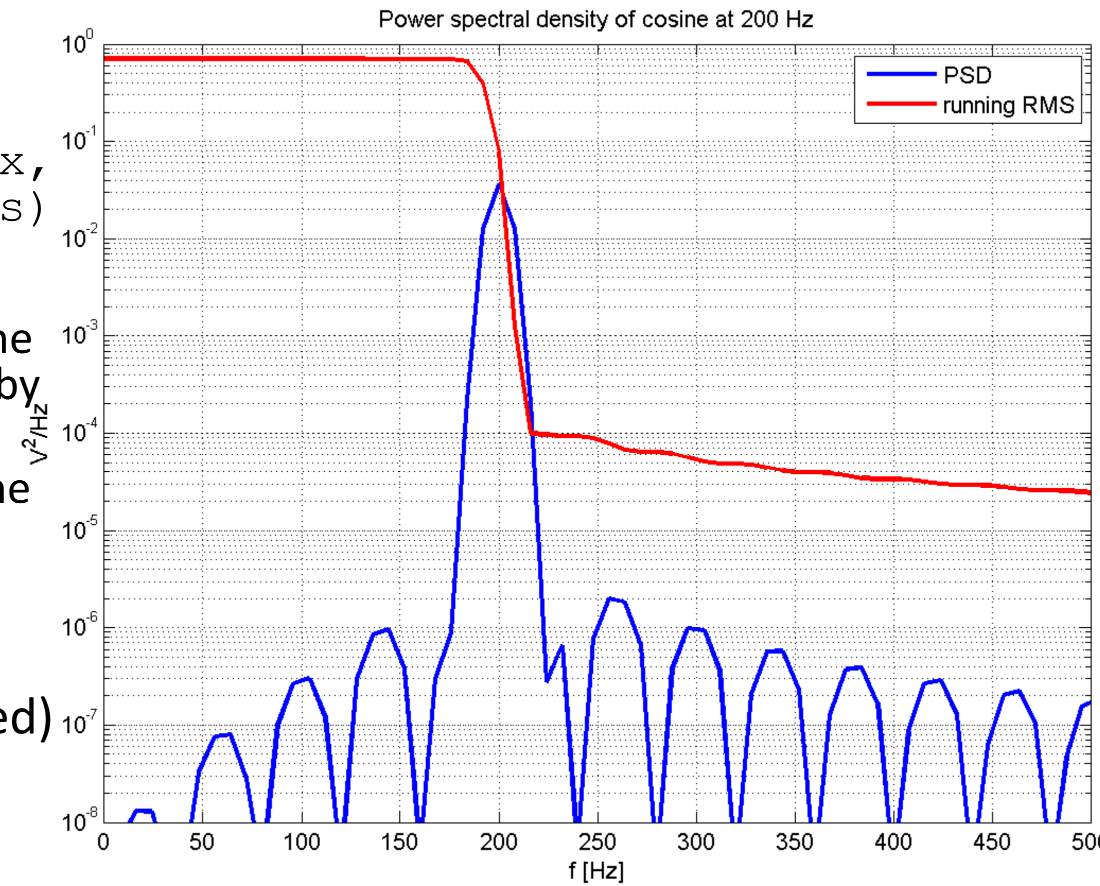
spectraldensity_example.m

Power Spectral Density (PSD)

- In this example, power is computed using
  - `w=hamming(length(x))`
  - `[Pxx,f]=periodogram(x, wi,'onesided',NFFT,Fs)`
- Data windowing
  - In the fft process, power in one frequency bin "leaks" to nearby bins.
  - Filter (with a window filter) the input data stream
- The RMS of a sinusiod: $1/\sqrt{2}$
- The (running) RMS computed using the PSD (and shown in red)

$$RMS = \sqrt{\sum P_{xx} \cdot \Delta f}$$

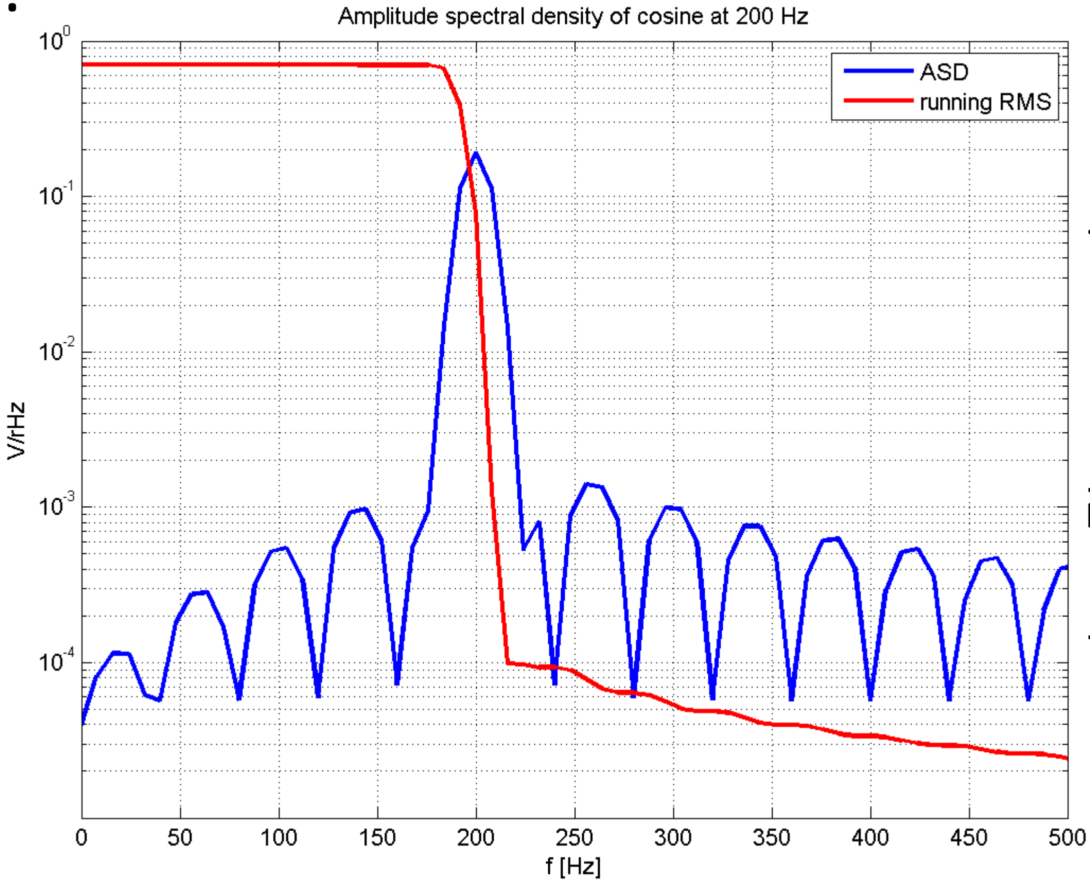
- The computed RMS agrees with the theoretical $1/\sqrt{2}$



spectraldensity_example.m

Amplitude Spectral Density (ASD)

- Plotting the amplitude:
  - simply the square root of the power spectral density $\sqrt{P_{xx}}$



spectraldensity_example.m

Finally – the question on sampling

The Discrete-time Fourier Transform (DTFT) was defined as

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)\, e^{-j\omega n}$$

where digital frequency ω

$$\omega = s\, T_s = 2\pi f \cdot T_s$$

and sampling frequency F_s

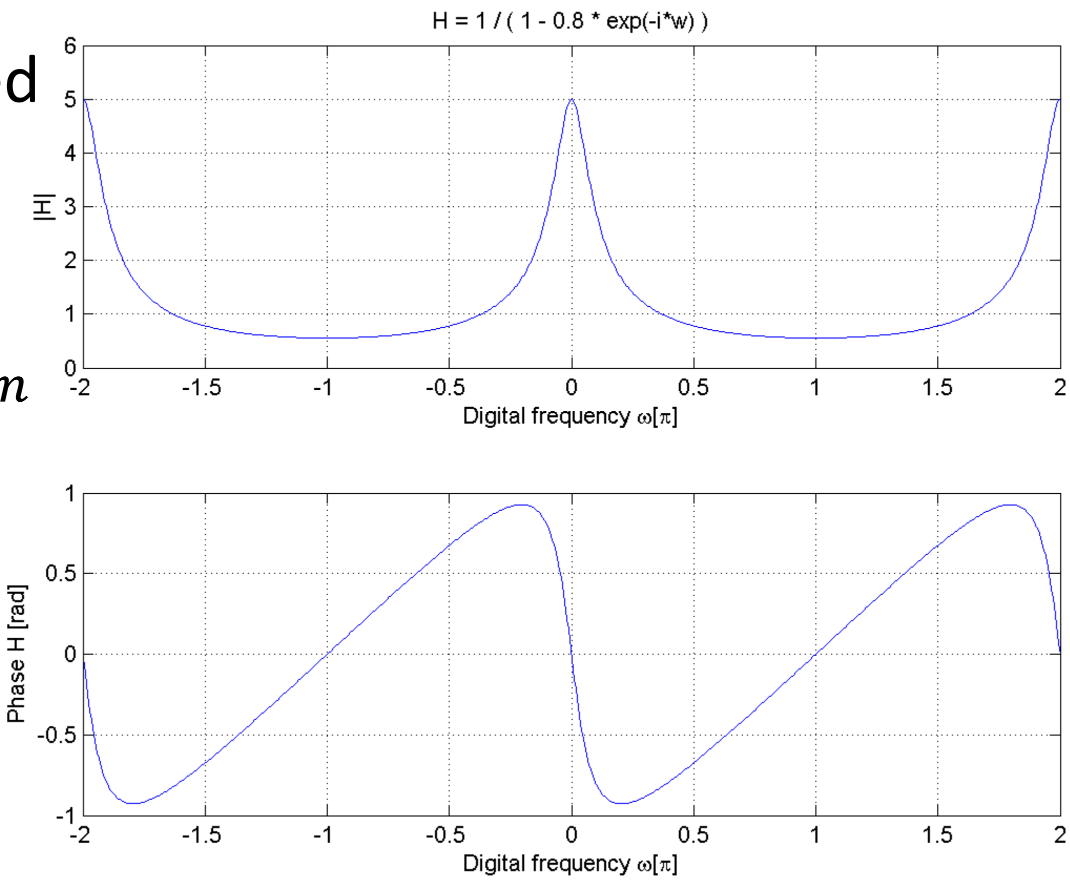
$$F_s = \frac{1}{T_s}$$

Finally – the question about sampling

The Discrete-time Fourier Transform (DTFT) is defined as

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)\, e^{-j\omega n}$$

where ω is the digital frequency.



If you inspect previous plots of DTFTs, you notice a periodicity of 2π.

Finally – the question about sampling

The $x(n)$ sequence represents a continuous-time signal $x_a(t)$ sampled every T_s seconds:

$$x(n) = x_a(nT_s)$$

where the digital frequency ω is

$$\omega = 2\pi\, f\, T_s$$

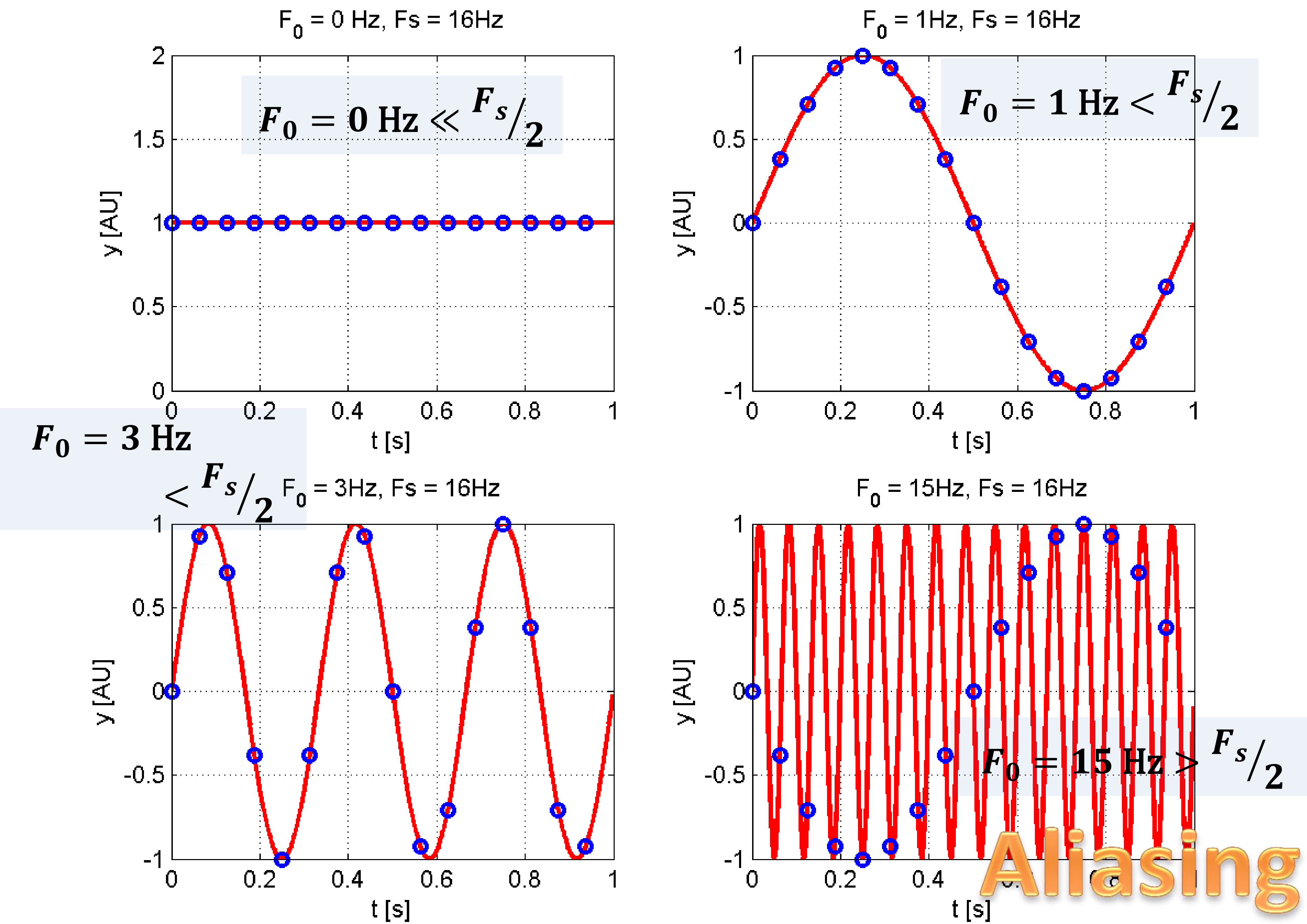
and f is frequency in Hz.

Defining the sampling frequency $F_s = 1/T_s$, the periodicity is

$$\omega = 2\pi\, f\, T_s = 2\pi \\ \rightarrow f = F_s$$

The signal repeats every F_s Hz.

$F_s/2$ is defined as the Nyquist frequency.



F_0 = 0 Hz, Fs = 16Hz
$F_0 = 0\ \text{Hz} \ll F_s/2$
F_0 = 1Hz, Fs = 16Hz
$F_0 = 1\ \text{Hz} < F_s/2$
$F_0 = 3\ \text{Hz} < F_s/2$
F_0 = 3Hz, Fs = 16Hz
F_0 = 15Hz, Fs = 16Hz
$F_0 = 15\ \text{Hz} > F_s/2$
Aliasing
y [AU]
t [s]

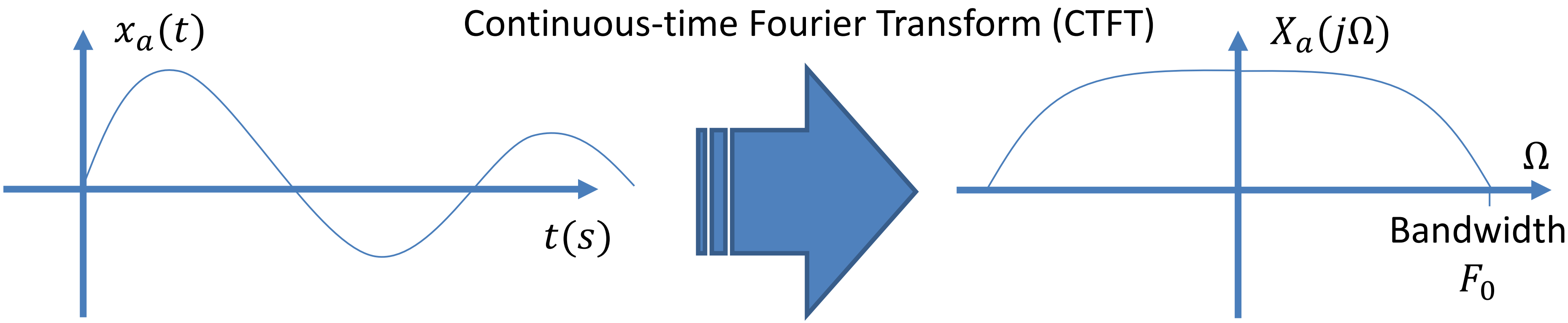
Sampling Principle

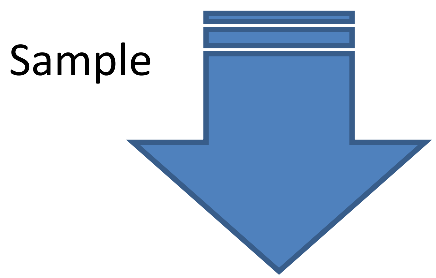
A band-limited signal $x_a(t)$ with bandwidth F_0 can be reconstructed from its sample values $x(n) = x_a(n\,T_s)$ if the bandwidth F_0 is less than the Nyquist frequency $F_n = F_s/2$

$$F_0 < F_n$$

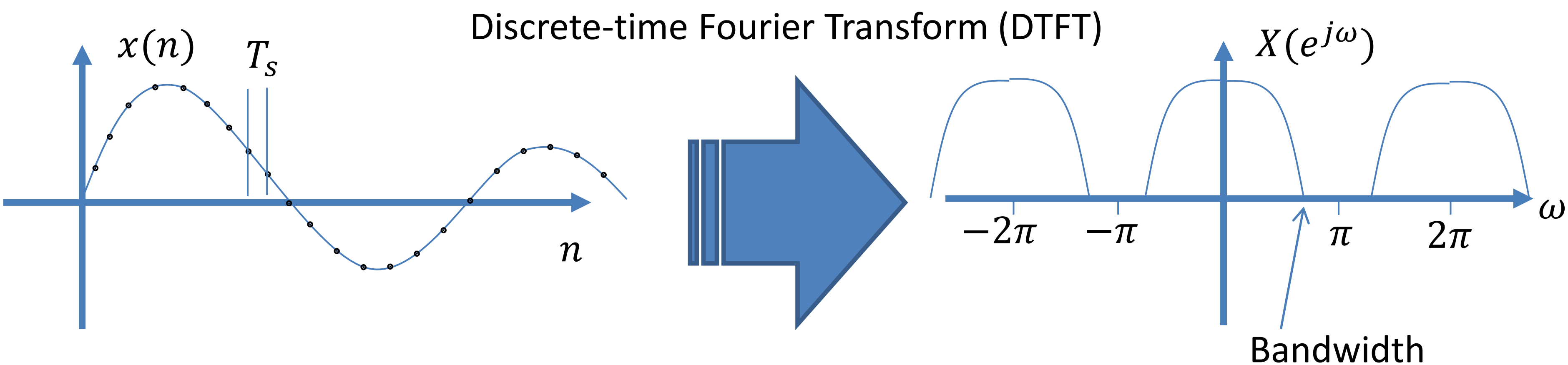
Otherwise aliasing (or distortion) would result in the reconstruction of $x_a(t)$.

Effect of sampling

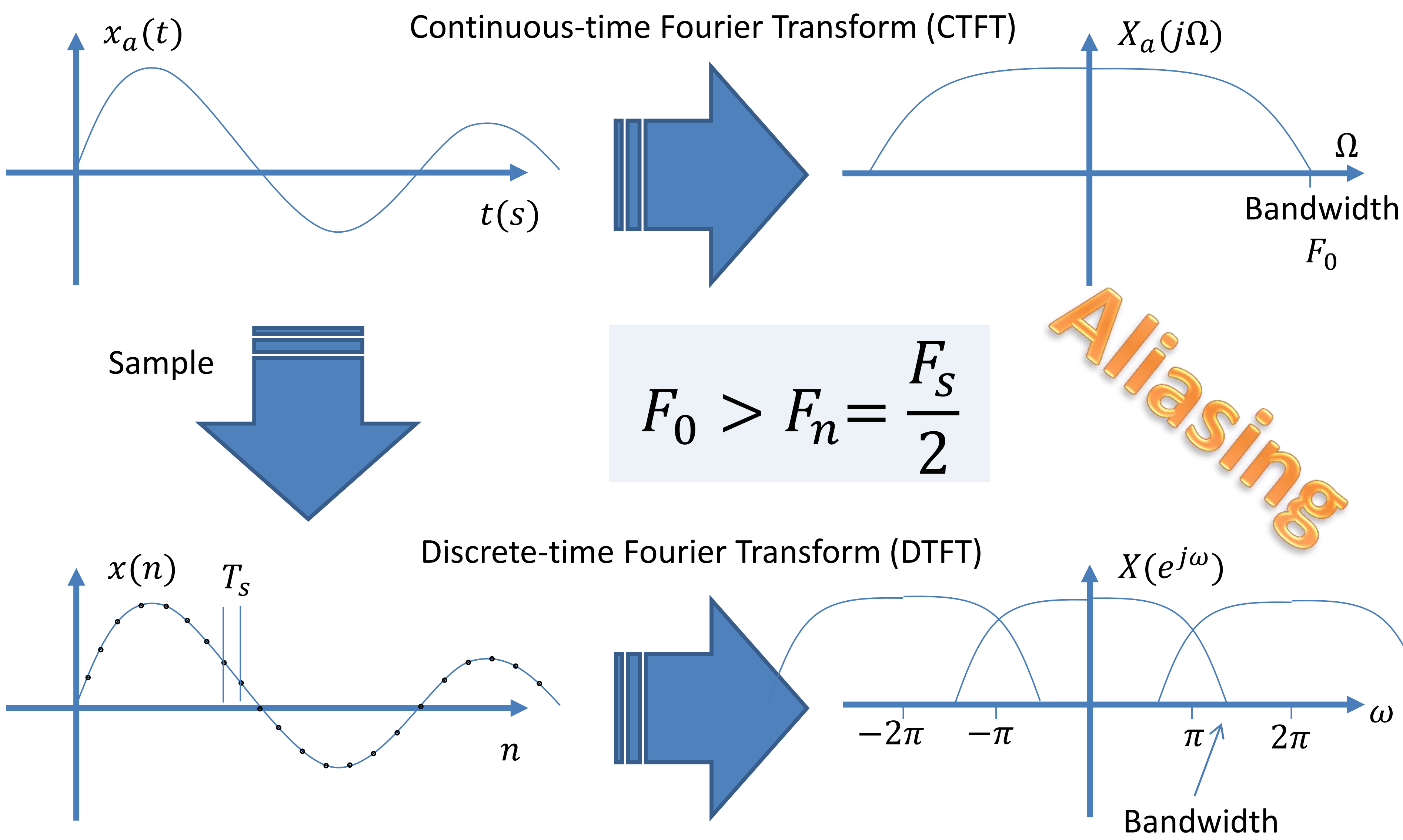




$$F_0 < F_n = \frac{F_s}{2}$$



Effect of sampling



Let's simulate aliasing

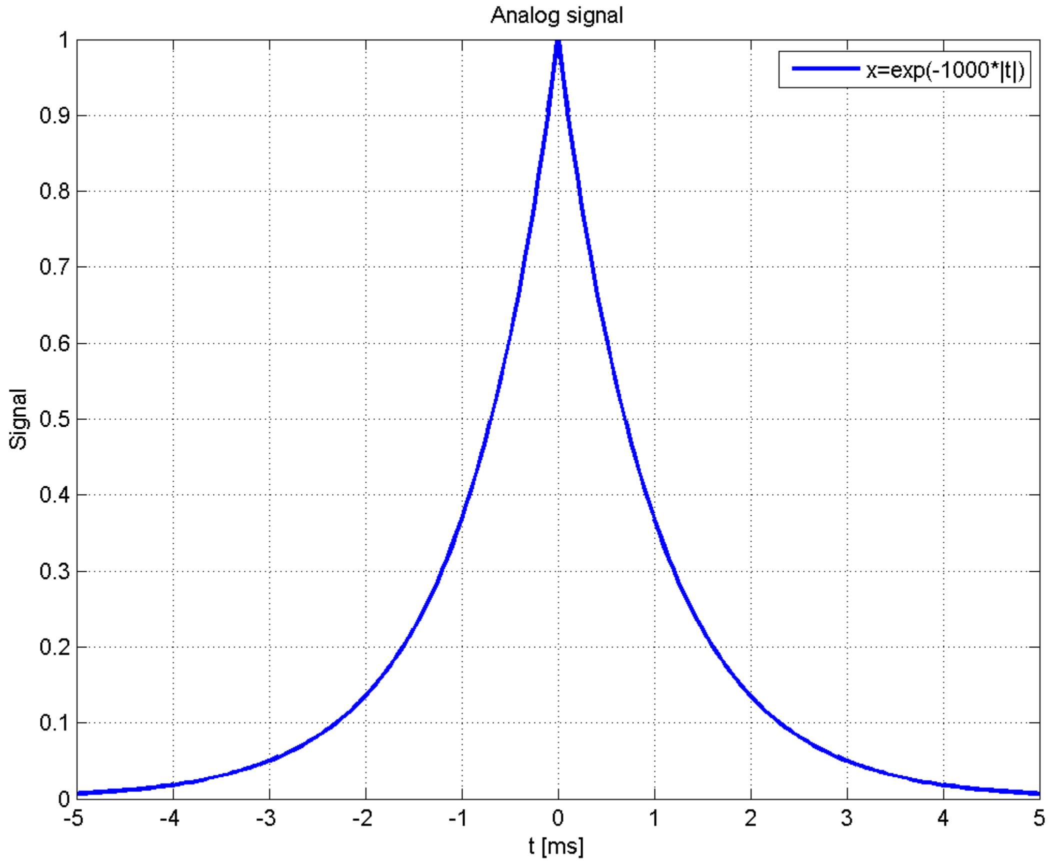
Suppose having an analog signal of the form

$$x_a(t) = e^{-1000\,|t|}$$

Its Fourier transform is

$$X_a(f) = \int_{-\infty}^{+\infty} x_a\,(t) e^{-i2\pi f t} dt = \frac{0.002}{1 + \left(\frac{2\pi f}{1000}\right)^2}$$

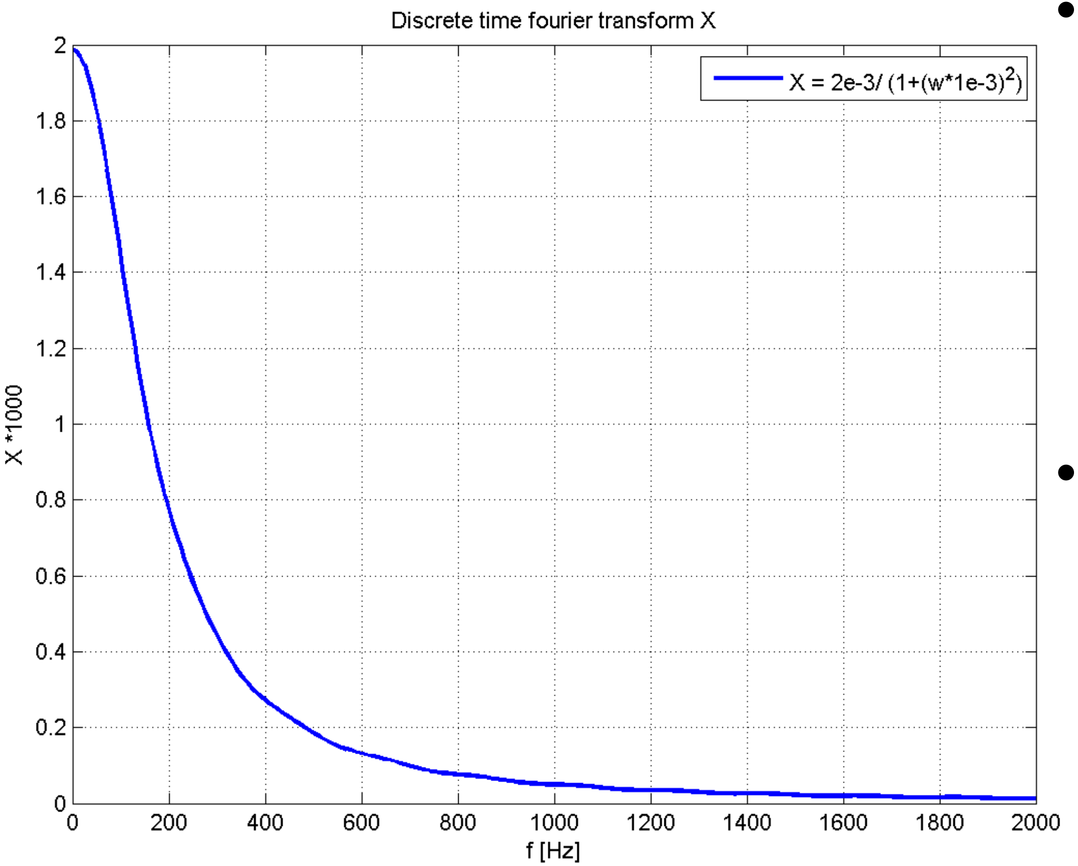
Aliasing



- Plot of $x_a(t)$ vs. time
- For $t > 5ms$ and $t < -5ms$
  - No significant energy left
  - Let's set $x_a = 0$

aliasing_example.m

Aliasing



- Plot of $X_a(f)$ vs. frequency
- For f > 2 kHz
  - No significant energy left
  - Set $X_a = 0$
  - Reasonable to set the signal's bandwidth to

$$F_0 = 2 \text{ kHz}$$

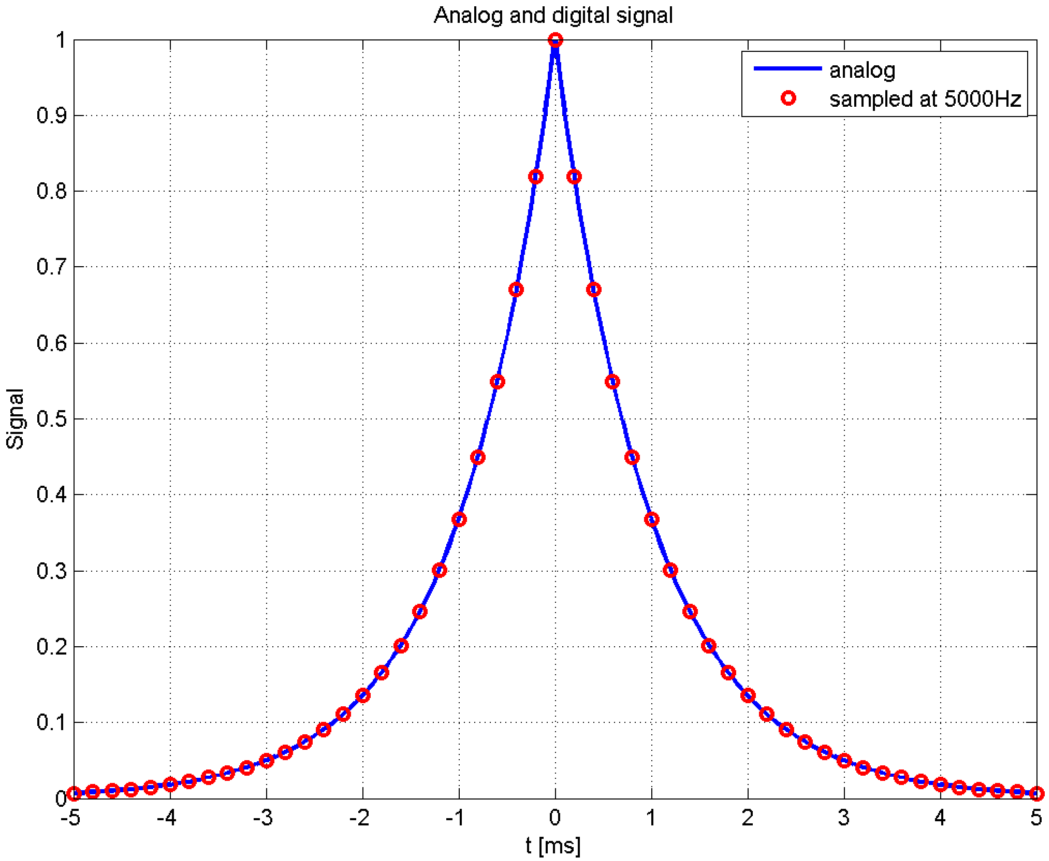
- To avoid aliasing the sampling frequency F_s must satisfy

$$F_0 < \frac{F_s}{2}$$

$$F_s > 2\,F_0 = 4 \text{ kHz}$$

aliasing_example.m

Aliasing



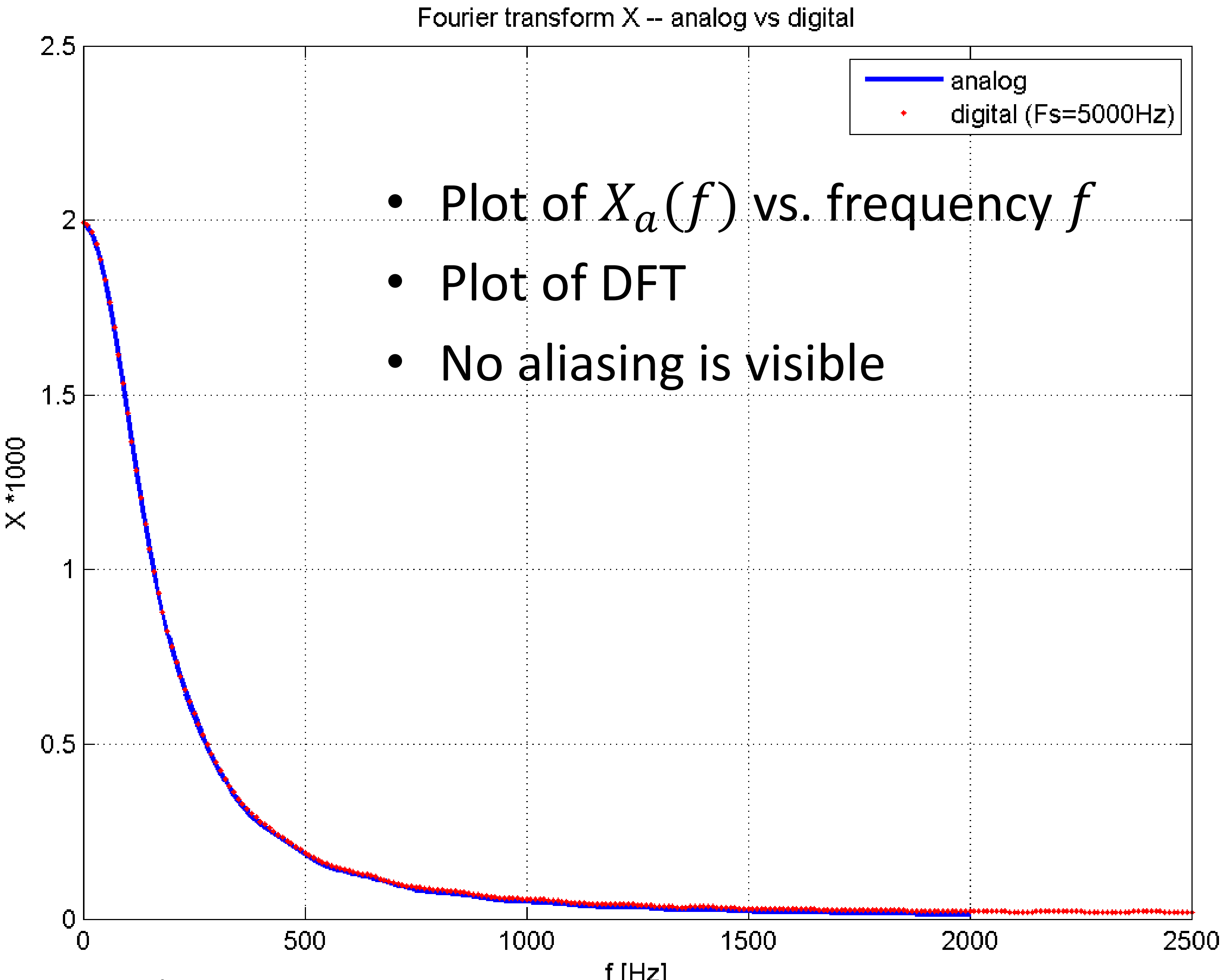
- Plot of x_a vs. time along with a sampled sequence ($F_s = 5$ kHz)
- According to the sampling principle

$$F_s = 5\,kHz > 2\,F_0 = 4\text{ kHz}$$

Should not have aliasing…

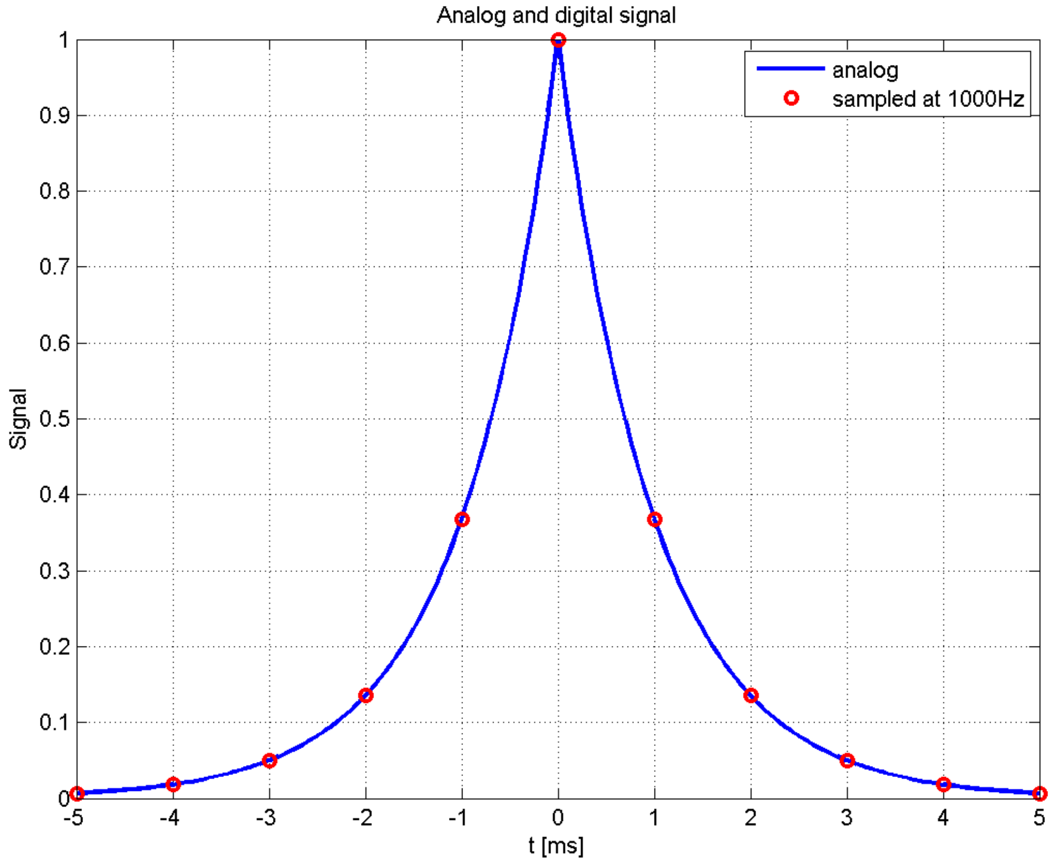
aliasing_example.m

- Plot of $X_a(f)$ vs. frequency f
- Plot of DFT
- No aliasing is visible



aliasing_example.m

Aliasing

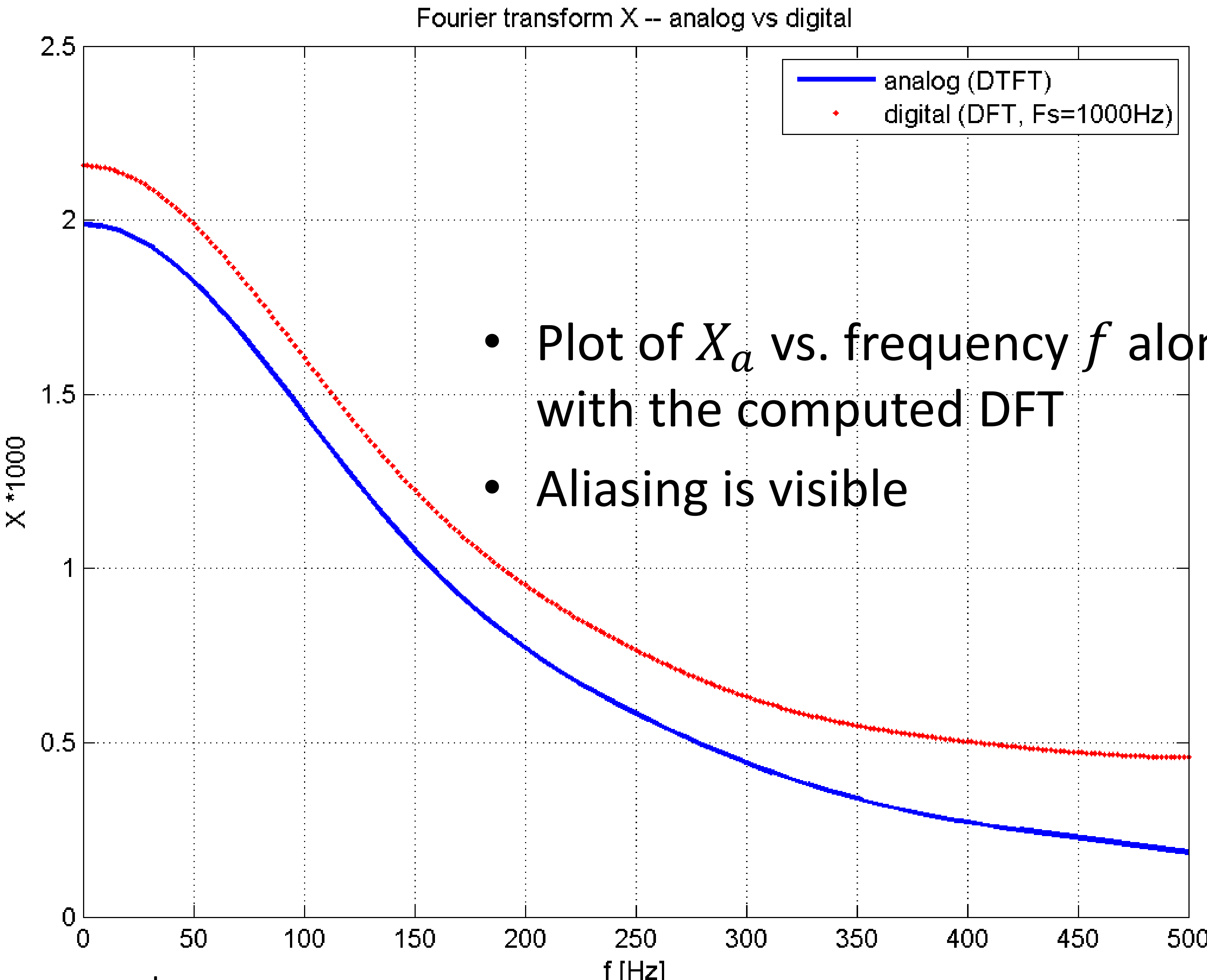


- Plot of x_a vs. time along with a sampled sequence ($F_s = 1$ kHz)
- According to the sampling principle

$$F_s = 1\,kHz \ngtr 2\,F_0 = 4\text{ kHz}$$

- Should have aliasing…

aliasing_example.m



- Plot of X_a vs. frequency f along with the computed DFT
- Aliasing is visible

aliasing_example.m

Recap

- The Discrete-time Fourier Transform (DTFT)
  - A very different but very useful representation of a sequence or system
  - Mapping into frequency space
  - $z = e^{j\omega}$ in the $\mathcal{Z}$ transform
- The Discrete Fourier Transform (DFT)
  - Obtained by sampling the DTFT in the frequency domain
  - FFT: Fast Fourier Transform
    - Algorithm for the efficient computation of DFTs
- Power Spectral Density
  - power of a signal over a particular frequency band.
- Sampling principle
  - The signal’s bandwidth F_0 must be less than the Nyquist frequency $F_n = F_s/2$ in order to avoid aliasing
- Modeling Aliasing