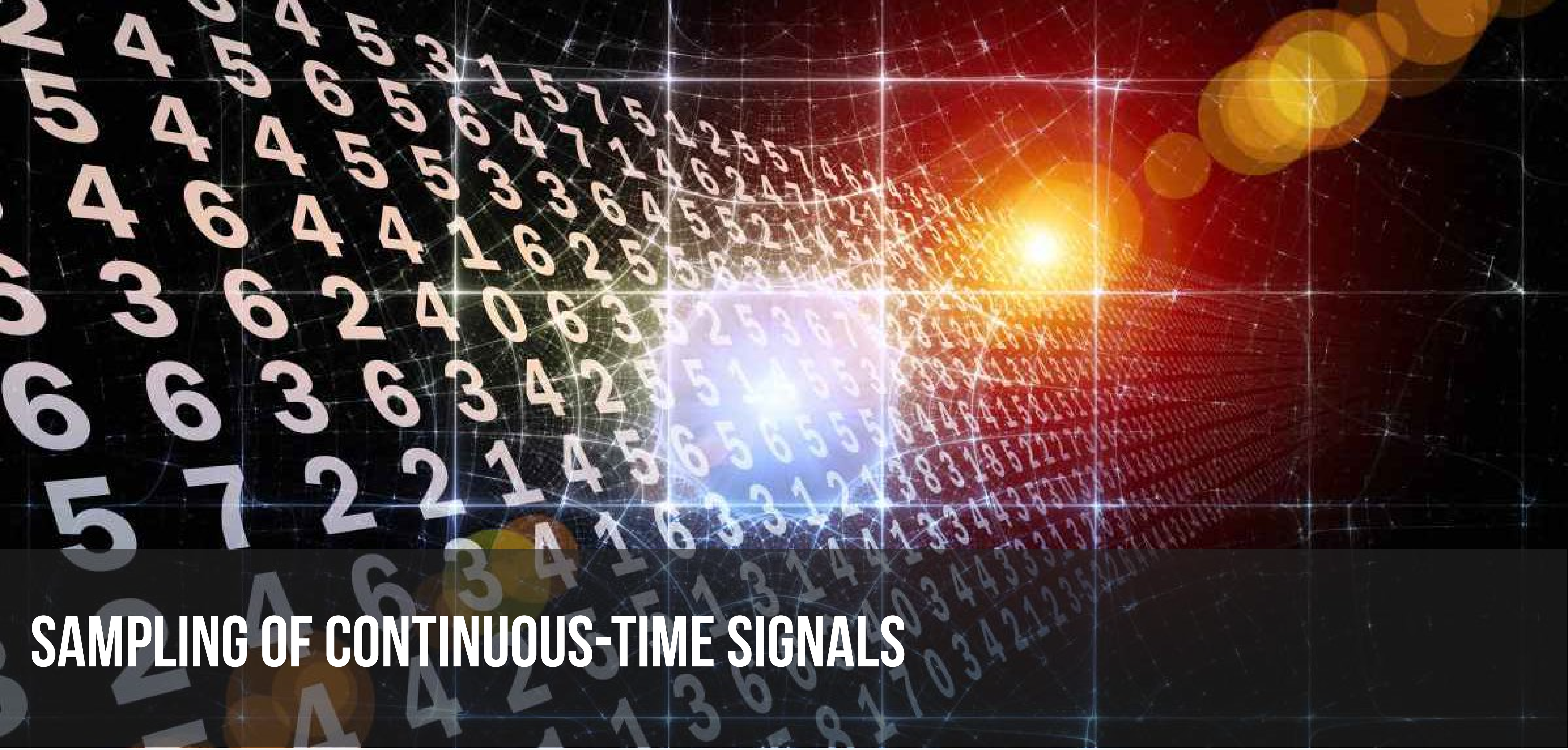
PENGOLAHAN SINYAL DIGITAL

Adhi Harmoko Saputro

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia



SAMPLING OF CONTINUOUS-TIME SIGNALS

Adhi Harmoko Saputro

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia

SIGNAL TYPES

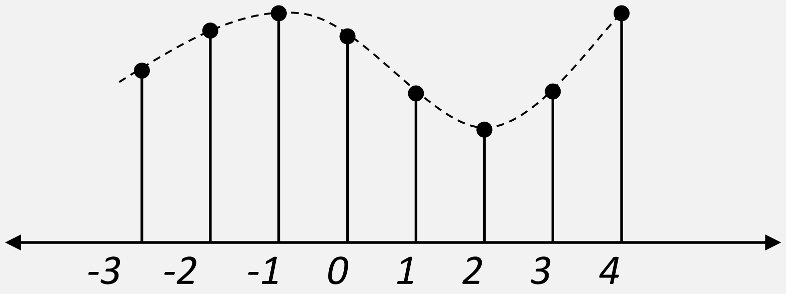
- Analog signals: continuous in time and amplitude
  - Example: voltage, current, temperature,…
- Digital signals: discrete both in time and amplitude
  - Example: attendance of this class, digitizes analog signals,…
- Discrete-time signal: discrete in time, continuous in amplitude
  - Example: hourly change of temperature in Jakarta

SIGNAL TYPES

- Theory for digital signals would be too complicated
  - Requires inclusion of nonlinearities into theory
- Theory is based on discrete-time continuous-amplitude signals
  - Most convenient to develop theory
  - Good enough approximation to practice with some care
- In practice we mostly process digital signals on processors
  - Need to take into account finite precision effects
- Our text book is about the theory hence its title
  - Discrete-Time Signal Processing

PERIODIC (UNIFORM) SAMPLING

- Sampling is a continuous to discrete-time conversion



- Most common sampling is periodic

$$x[n] = x_c(nT) \quad -\infty < n < \infty$$

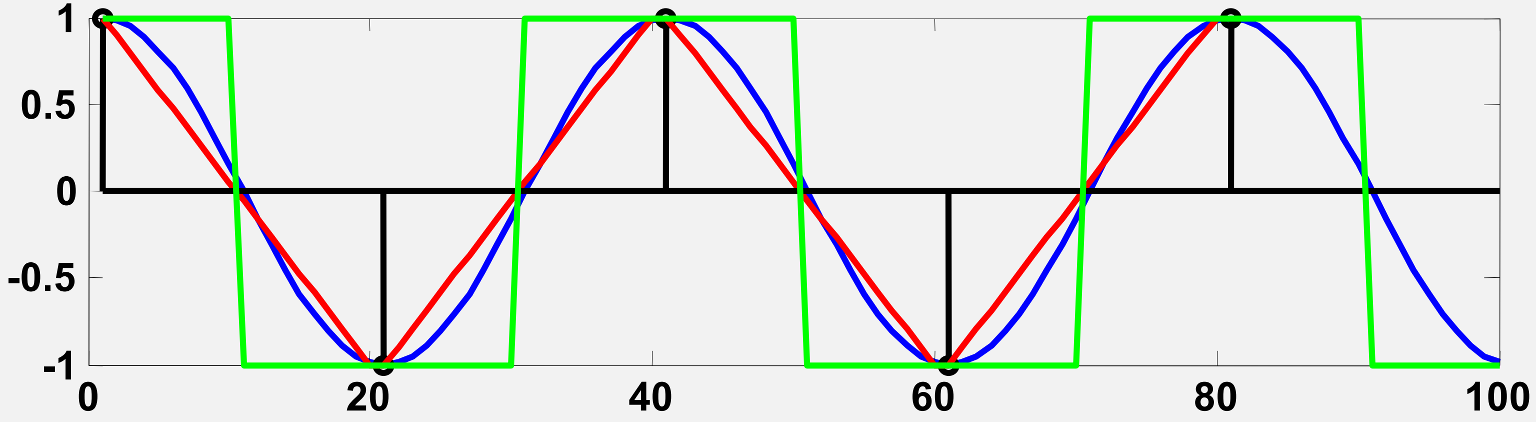
- T is the sampling period in second
- f_s = 1/T is the sampling frequency in Hz

PERIODIC (UNIFORM) SAMPLING

- Sampling frequency in radian-per-second $\Omega_s = 2\pi f_s$ rad/sec
- Use [.] for discrete-time and (.) for continuous time signals
- This is the ideal case not the practical but close enough
  - In practice it is implement with an analog-to-digital converters
  - We get digital signals that are quantized in amplitude and time

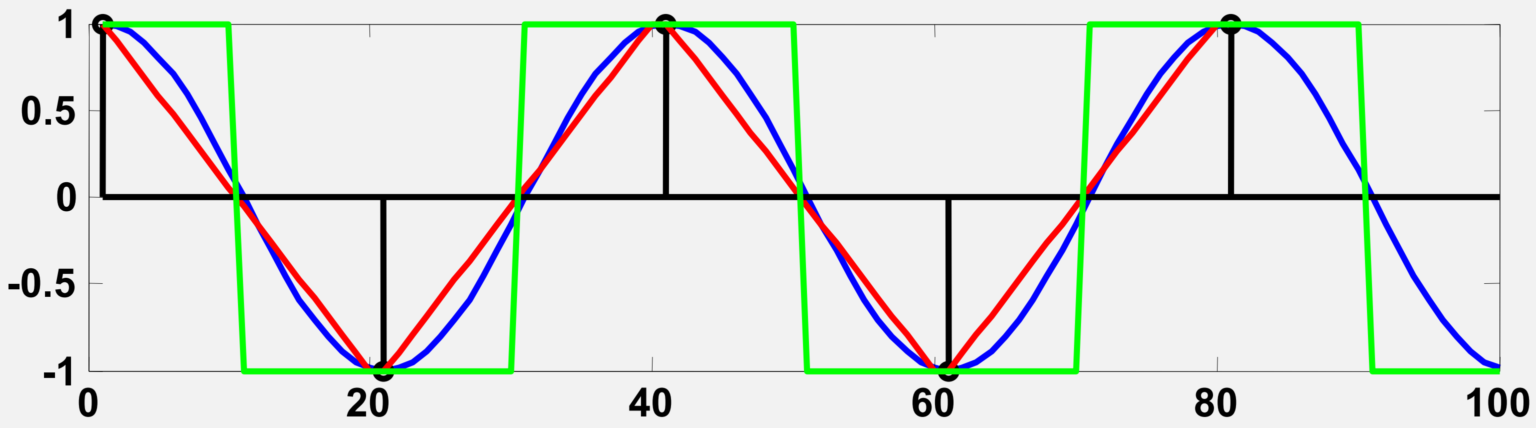
PERIODIC SAMPLING

- Sampling is, in general, not reversible
- Given a sampled signal one could fit infinite continuous signals through the samples



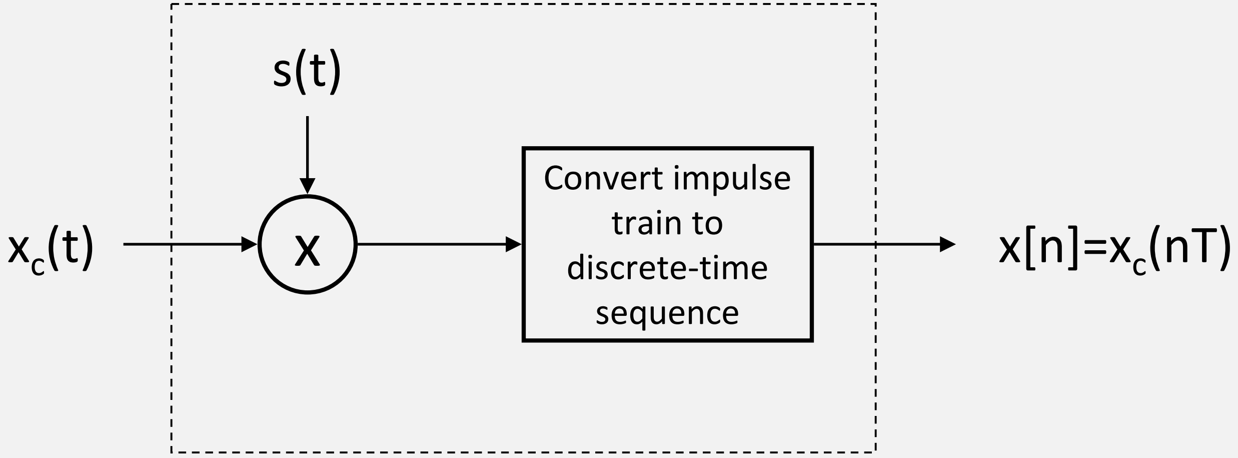
PERIODIC SAMPLING

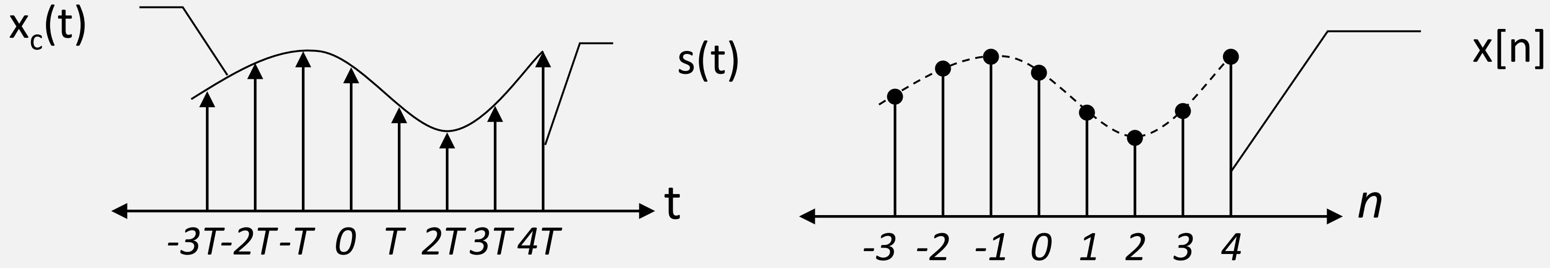
- Fundamental issue in digital signal processing
  - If we loose information during sampling we cannot recover it
- Under certain conditions an analog signal can be sampled without loss so that it can be reconstructed perfectly

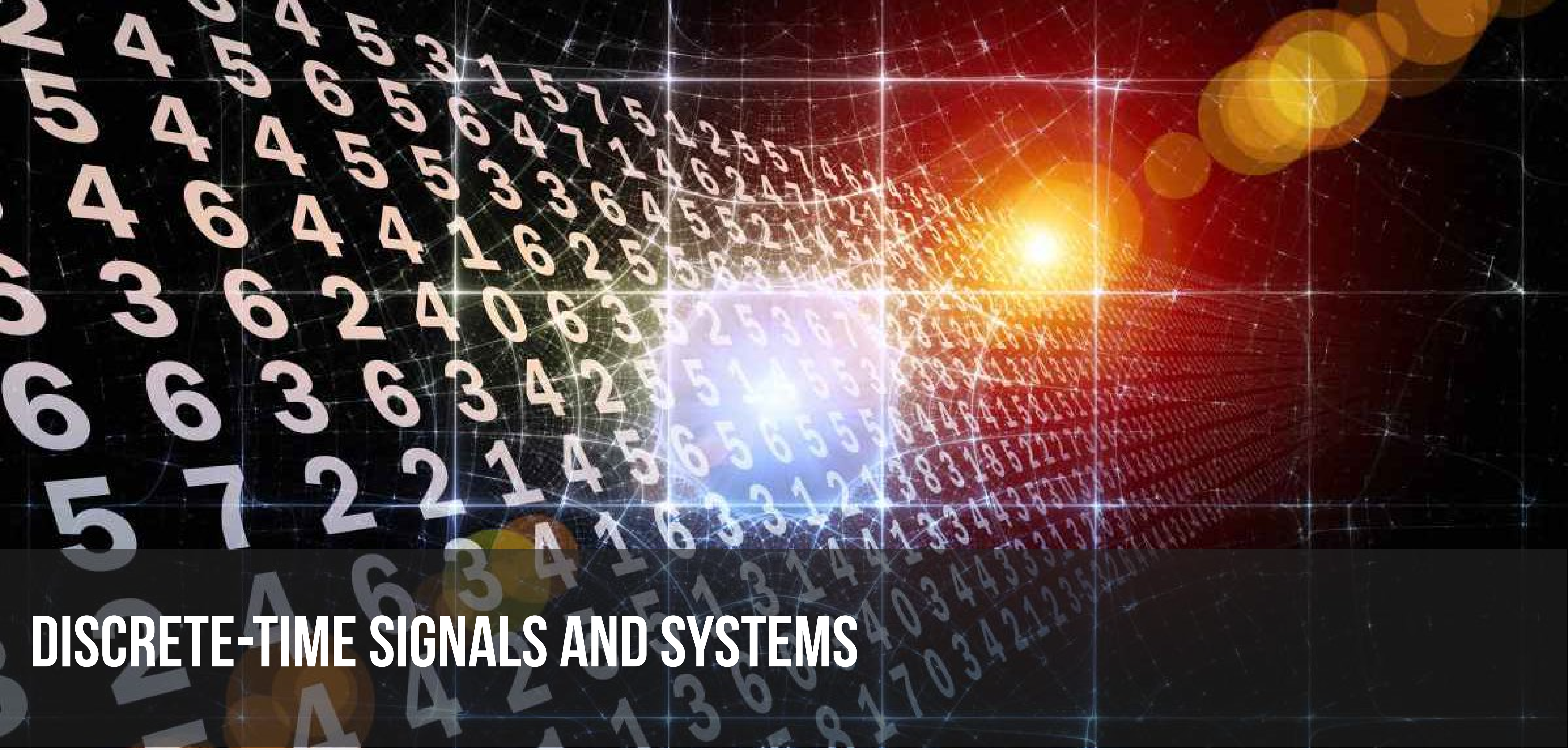


REPRESENTATION OF SAMPLING

- Mathematically convenient to represent in two stages
  - Impulse train modulator
  - Conversion of impulse train to a sequence







DISCRETE-TIME SIGNALS AND SYSTEMS

Adhi Harmoko Saputro

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia

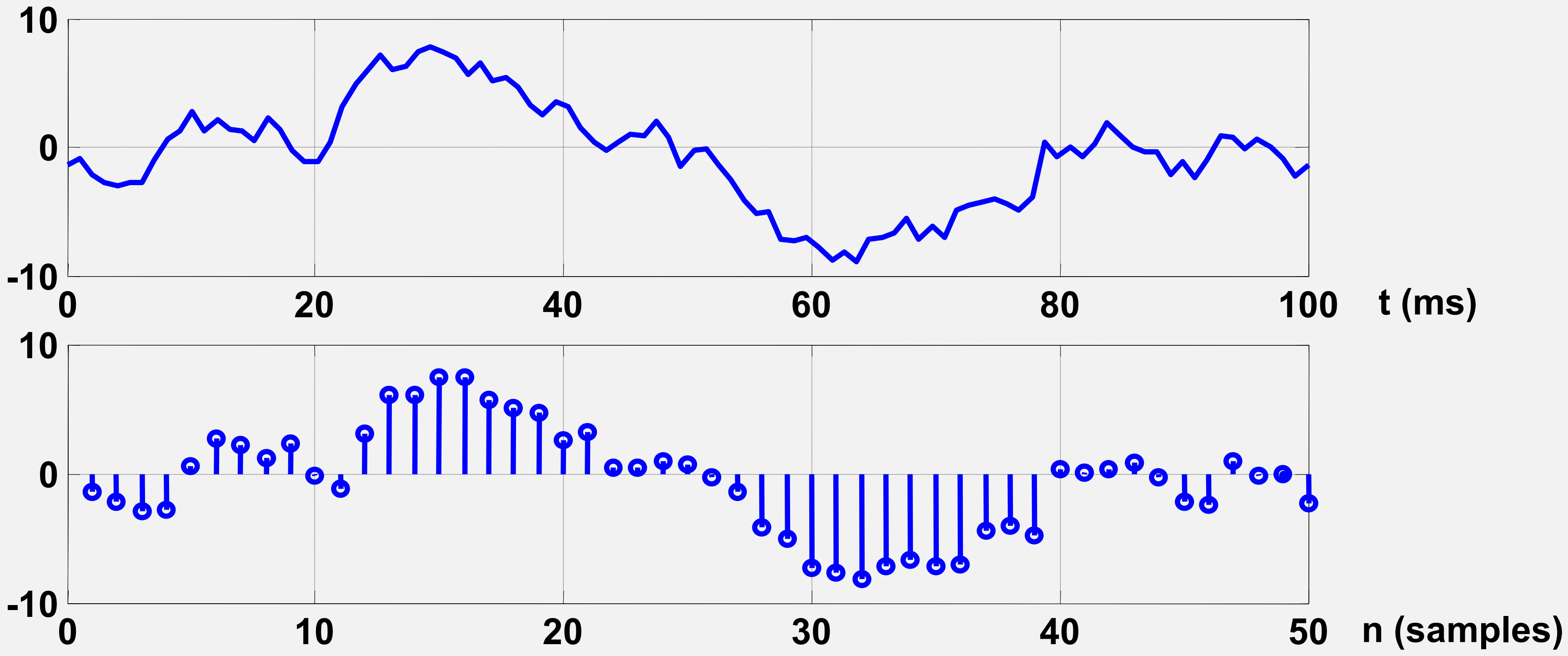
DISCRETE-TIME SIGNALS: SEQUENCES

- Discrete-time signals are represented by sequence of numbers
  - The n^{th} number in the sequence is represented with x[n]

$$x[n] = \{x[n]\} = \{\ldots, x[-1], x[0], x[1], \ldots\}$$

- Often times sequences are obtained by sampling of continuous-time signals
  - In this case x[n] is value of the analog signal at $x_c(nT)$
  - Where T is the sampling period

DISCRETE-TIME SIGNALS: SEQUENCES



DISCRETE-TIME SIGNALS: SEQUENCES

- In MATLAB we can represent a *finite-duration* sequence by a *row vector* of appropriate values
- A vector does not have any information about sample position *n*.
  - A correct representation of $x(n)$ would require two vectors, one each for *x* and *n*.

$$x[n] = \left\{2, 1, -1, \underset{\uparrow}{0}, 1, 4, 3, 7\right\}$$

  - represented in MATLAB

```
>> n=[-3,-2,-1,0,1,2,3,4]; x=[2,1,-1,0,1,4,3,7];
```

BASIC SEQUENCES AND OPERATIONS

- Delaying (Shifting) a sequence

$$y[n] = x[n - n_o]$$

- Unit sample (impulse) sequence

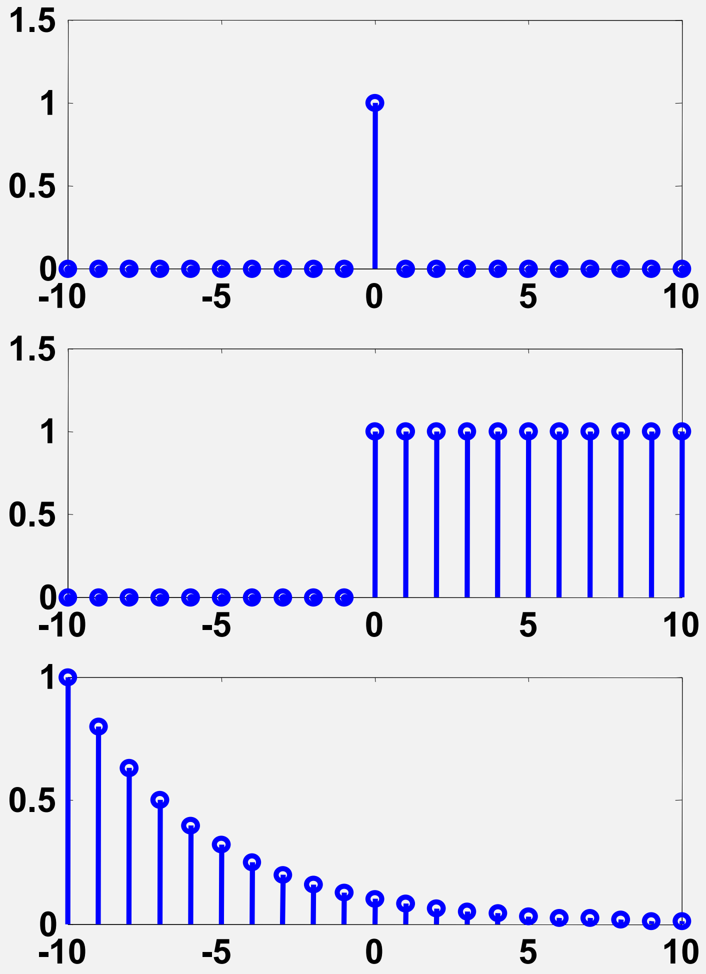
$$\delta[n] = \begin{cases} 0 & n \neq 0 \\ 1 & n = 0 \end{cases}$$

- Unit step sequence

$$u[n] = \begin{cases} 0 & n < 0 \\ 1 & n \geq 0 \end{cases}$$

- Exponential sequences

$$x[n] = A\alpha^n$$



SINUSOIDAL SEQUENCES

- Important class of sequences

$$x[n] = \cos(\omega_o n + \varphi)$$

- An exponential sequence with complex

$$\alpha = |\alpha| e^{j\omega_o} \text{ and } A = |A| e^{j\varphi}$$

$$x[n] = A\alpha^n = |A| e^{j\varphi} |\alpha|^n e^{j\omega_o n} = |A||\alpha|^n e^{j(\omega_o n + \varphi)}$$

$$x[n] = |A||\alpha|^n \cos(\omega_o n + \varphi) + j|A||\alpha|^n \sin(\omega_o n + \varphi)$$

- x[n] is a sum of weighted sinusoids

SINUSOIDAL SEQUENCES

- Different from continuous-time, discrete-time sinusoids
  - Have ambiguity of $2\pi k$ in frequency

$$\cos\left(\left(\omega_o + 2\pi k\right)n + \varphi\right) = \cos\left(\omega_o n + \varphi\right)$$

  - Are not necessary periodic with $2\pi/\omega_o$

$$\cos\left(\omega_o n + \varphi\right) = \cos\left(\omega_o n + \omega_o N + \varphi\right) \text{ only if } N = \frac{2\pi k}{\omega_o} \text{ is an integer}$$

MATLAB EXAMPLE 1

- Create sequence

$$\delta[n-n_o]=\begin{cases}0 & n\neq 0\\ 1 & n=0\end{cases}$$

- over the $0 \le n \le 10$ interval, with $n_o = 5$

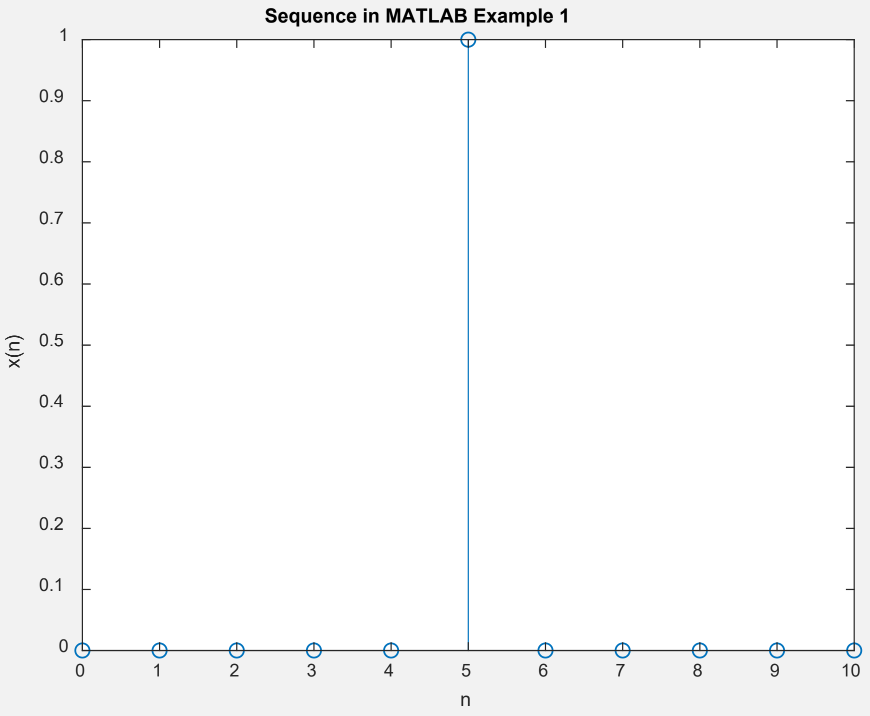
```
n0 = 5;
n1 = 0;
n2 = 10;
n = n1:n2;
x = (n-n0) == 0; stem(n, x);
```

MATLAB EXAMPLE 1

- Create sequence

$$\delta[n-n_o]=\begin{cases}0 & n\neq 0\\ 1 & n=0\end{cases}$$

- over the $0 \le n \le 10$ interval, with $n_o = 5$



MATLAB EXAMPLE 2

- Create a function for following sequence

$$\delta[n-n_o]=\begin{cases}0 & n\neq 0\\ 1 & n=0\end{cases}$$

- over the $n_1 \leq n_0 \leq n_2$ interval

```
function [x,n] = impseq(n0,n1,n2)
% Generates x(n) = delta(n-n0); n1 <= n <= n2
% ----------------------------------------------
% [x,n] = impseq(n0,n1,n2)
%
n = n1:n2; x = (n-n0) == 0;
```

```
n0 = 5;
n1 = 0;
n2 = 10;
[x,n] = impseq(n0,n1,n2); stem(n, x);
```

MATLAB EXAMPLE 3

- Create sequence

$$u[n-n_o]=\begin{cases}0 & n<0\\ 1 & n\geq 0\end{cases}$$

- over the *$0 \leq n \leq 10$* interval, with $n_o = 5$

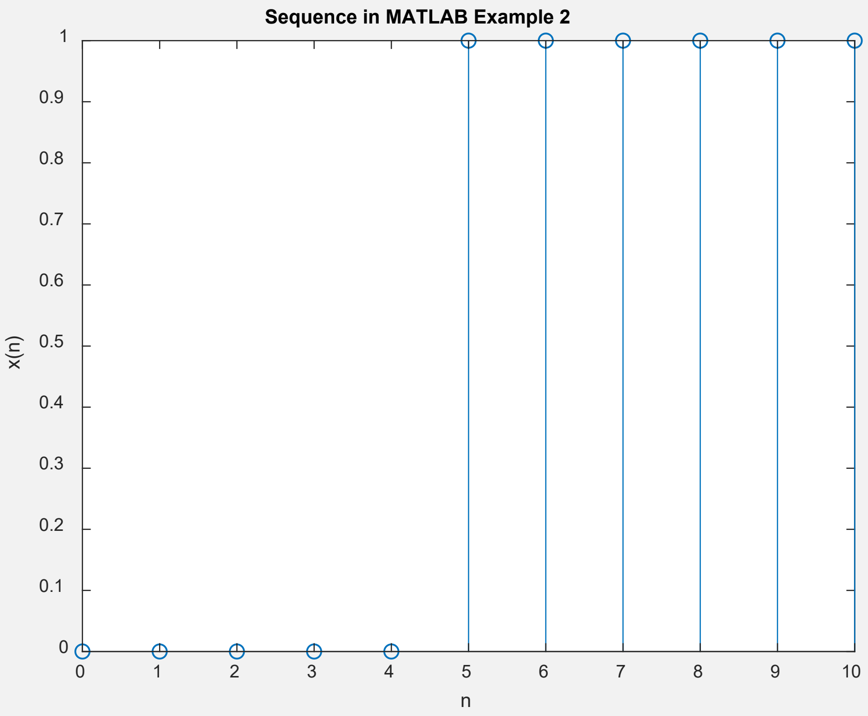
```
n0 = 5;
n1 = 0;
n2 = 10;

n = n1:n2; x = (n-n0) >= 0;
stem(n, x);
```

MATLAB EXAMPLE 3

- Create sequence

$$u[n-n_o]=\begin{cases}0 & n<0\\ 1 & n\geq 0\end{cases}$$

- over the $0 \le n \le 10$ interval, with $n_o = 5$



MATLAB EXAMPLE 2_1

- Create a function for following sequence

$$u[n-n_o]=\begin{cases}0 & n<0\\ 1 & n\geq 0\end{cases}$$

- over the $n_1 \leq n_0 \leq n_2$ interval

```
function [x,n] = stepseq(n0,n1,n2)
% Generates x(n) = u(n-n0); n1 <= n <= n2
% ------------------------------------------
% [x,n] = stepseq(n0,n1,n2)
%
n = n1:n2; x = (n-n0) >= 0;
```

```
n0 = 5;
n1 = 0;
n2 = 10;
[x,n] = stepseq(n0,n1,n2); stem(n, x);
```

MATLAB EXAMPLE 3

- Create sequence

$$x[n]=(0.9)^n$$

- over the $0 \le n \le 10$ interval

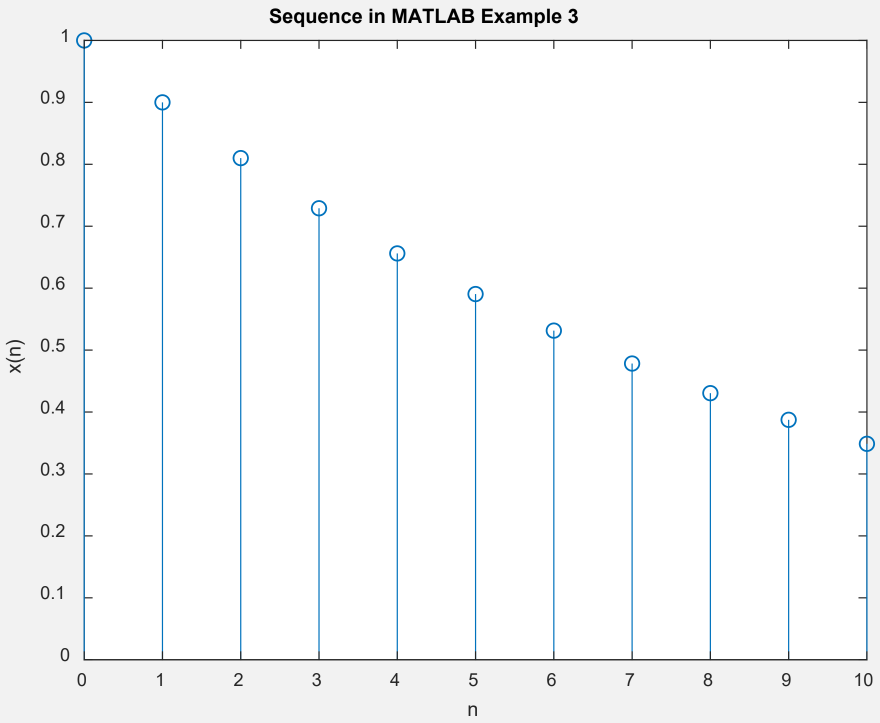
```
n = 0:10;
x = (0.9).^n;

stem(n,x);
```

MATLAB EXAMPLE 3

- Create sequence

$$x[n]=(0.9)^n$$

- over the $0 \le n \le 10$ interval



MATLAB EXAMPLE 4

- Create sequence

$$x[n] = \exp[(2 + j3)n]$$

- over the $0 \le n \le 10$ interval

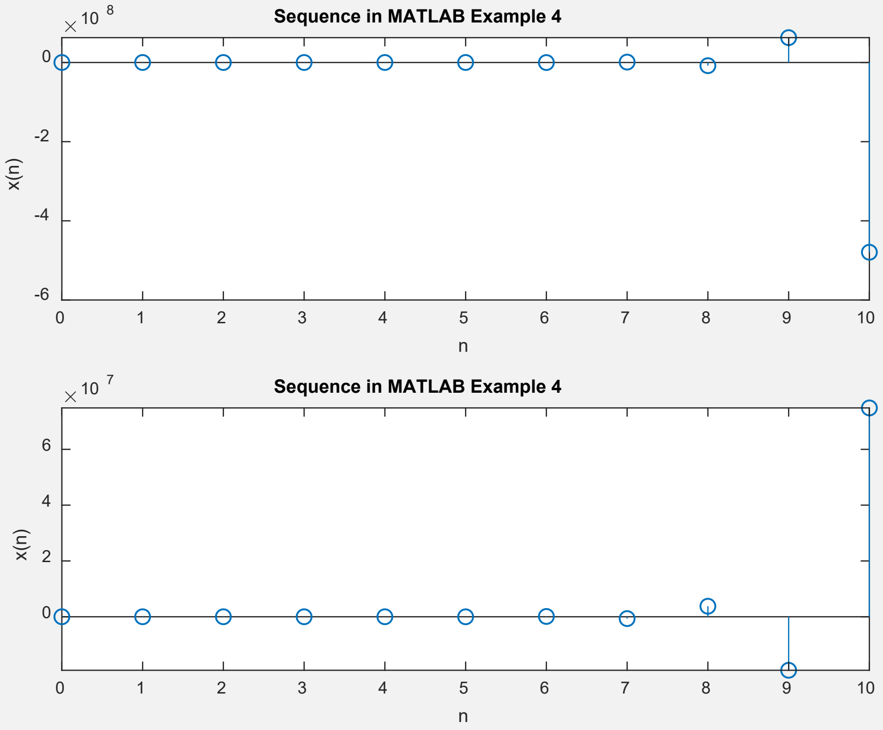
```
n = 0:10;
x = exp((2+3j)*n);

subplot(211); stem(n,imag(x));
subplot(212); stem(n,real(x));
```

MATLAB EXAMPLE 4

- Create sequence

$$x[n] = \exp[(2 + j3)n]$$

- over the $0 \leq n \leq 10$ interval



MATLAB EXAMPLE 5

- Create sequence

$$x[n] = 3\cos(0.1\pi n + \pi / 3) + 2\sin(0.5\pi n)$$

- over the $0 \le n \le 10$ interval

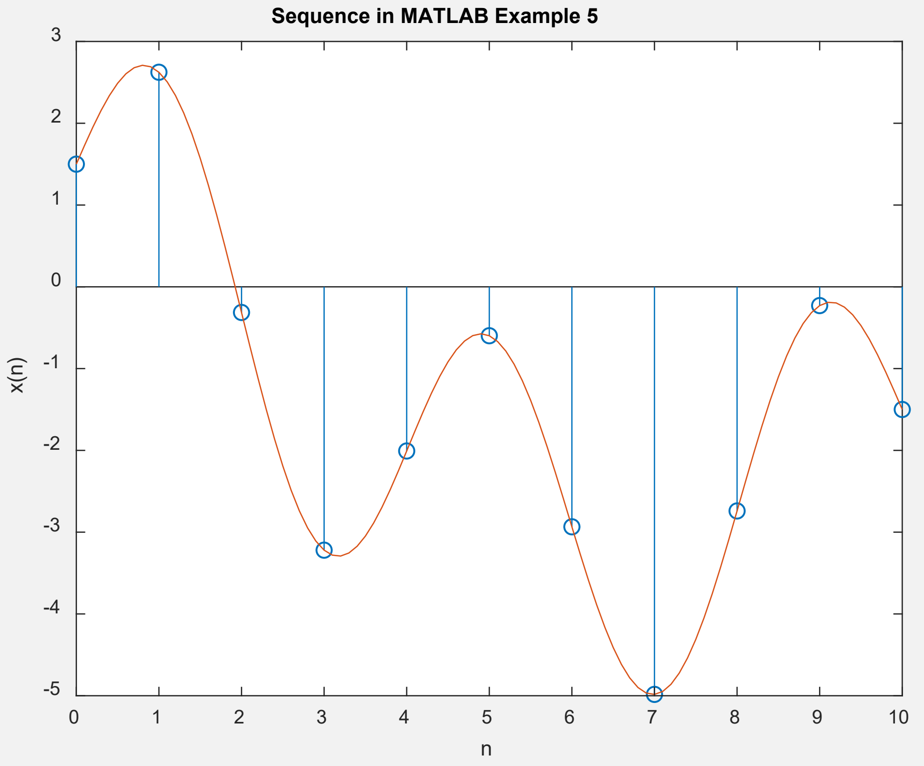
```
n = 0:10;
x = 3*cos(0.1*pi*n+pi/3)+2*sin(0.5*pi*n);

stem(n,x);
```

MATLAB EXAMPLE 5

- Create sequence

$$x[n] = 3\cos(0.1\pi n + \pi/3) + 2\sin(0.5\pi n)$$

- over the $0 \le n \le 10$ interval



TERIMA KASIH

Adhi Harmoko Saputro

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia