**ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)**
ORGANISATION OF ISLAMIC COOPERATION (OIC)
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

# EEE 4602: Signals & Systems Lab

# Lab – 04: Signal Processing

## 4.2 Signal Generation and Visualization

In this section, you will learn how to generate different waveforms using MATLAB's signal processing toolbox.

**Sine/Cosine wave**

You can generate periodic waveforms like sine or cosine wave in MATLAB using the following code –

```
x= -pi:0.001:2*pi
y= sin(x)
plot (x, y)
```

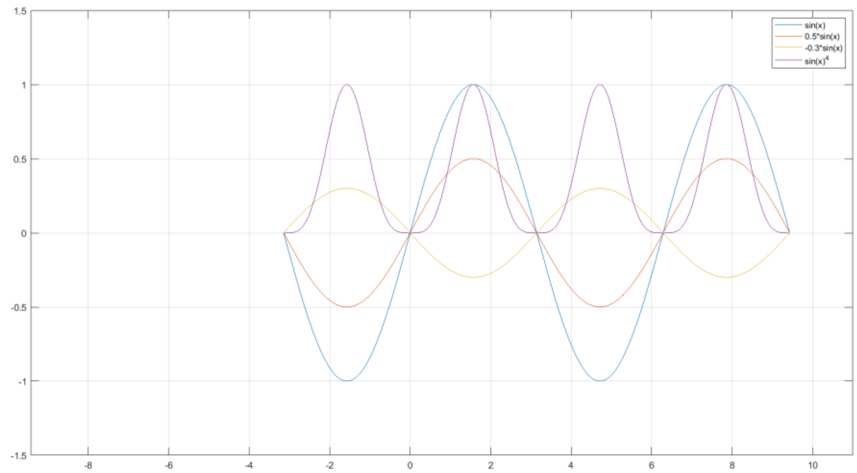
You can also change the amplitude of the signal by simply multiplying it by a factor.

```
x= -pi:0.001:3*pi
y= 0.5* abs(sin(x))
plot (x, y)
axis ([-3*pi, 3.5*pi, -1,1])
grid on
```

You can control the x/y - axis range using the axis function. You can also plot multiple waveforms in the same figure (Fig. 1).

```
x= -pi:0.001:3*pi
y= sin(x)
plot (x, y, x, 0.5*sin(x), x, -0.3*y, x, y.^4)
axis ([-3*pi, 3.5*pi, -1.5,1.5])
grid on
```

*Prepared by*
*Asif Newaz*
*Lecturer, EEE, IUT*



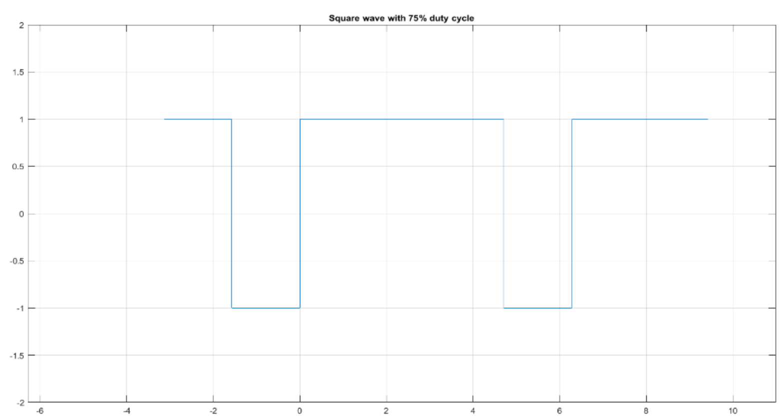

*1. Fig. 1*

## <u>Square Wave</u>

You can generate a square wave with a predefined duty cycle (default = 50%) using the following code –

```
x= -pi:0.001:3*pi
y= square (x, 75)
plot (x, y)
axis ([-2*pi, 3.5*pi, -2,2])
```

The 2nd parameter is the duty cycle with values ranging from 0 to 100.




*2. Fig. 2*

*Prepared by*
*Asif Newaz*
*Lecturer, EEE, IUT*

You can also generate signals with a specified frequency.

**Question:** Generate a 50 Hz square wave sampled at 1 kHz for 80 ms.

**Solution:**

```
x= 0:0.001:0.08
y= 2*pi*50*x
plot (x, square(y))
axis ([-0.01, 0.1, -2,2])
```

Here, the signal is to be displayed for 80 ms = 0.08 seconds.

Sample rate = 1KHz. Therefore, x is sampled with 1/1000 = 0.001 step size.

## Sawtooth Wave

You can generate a 50 Hz sawtooth wave using the following code –

```
f= 50
x= 0:0.001:.5
y= sawtooth(2*pi*f*x)
plot (x, y)
axis ([-0.01, 0.6, -2,2])
```

Defining the 2nd parameter of the 'sawtooth' function as 0.5 will produce a triangular wave.

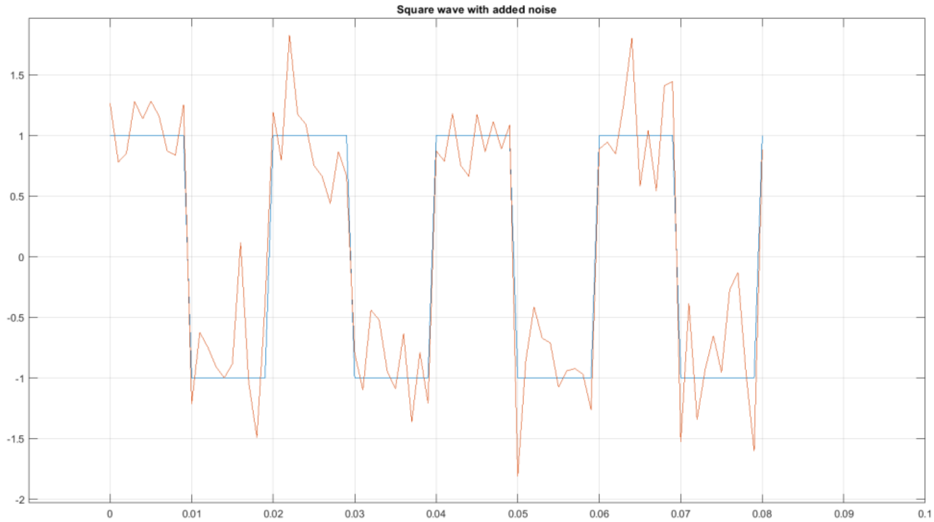
## Adding noise to the signal

The 'awgn' (Additive White Gaussian Noise) function adds white Gaussian noise to the signal. You can define the signal to noise ratio (snr) with the 2nd input parameter to the function. A value of 10 implies that the signal strength is 10 times higher than the noise.

```
x= 0:0.001:0.08
y= square(2*pi*50*x)
noise = awgn(y,10)

plot (x, y, x, noise)
axis ([-0.01, 0.1, -2,2])
```

*Prepared by*
*Asif Newaz*
*Lecturer, EEE, IUT*



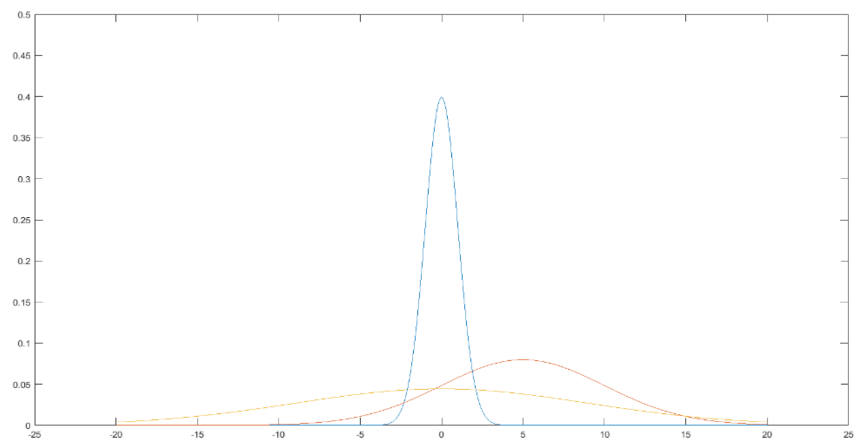

*3. Fig. 3*

## Gaussian Pulse

A gaussian pulse has a common bell-shaped curve. It has a shape that is similar to a Gaussian (normal) distribution. You can generate such a pulse using the following code –

```
t = -5:0.01:5;      % Time vector
mu = 0;             % Mean of the Gaussian pulse
sigma = 1;          % Standard deviation of the Gaussian pulse
gaussian_pulse = normpdf(t, mu, sigma);
```

Change the value of μ and σ and see how the waveform changes. Change in μ will shift the center while change in σ stretches the signal.



*4 Fig. 4*

*Prepared by*
*Asif Newaz*
*Lecturer, EEE, IUT*

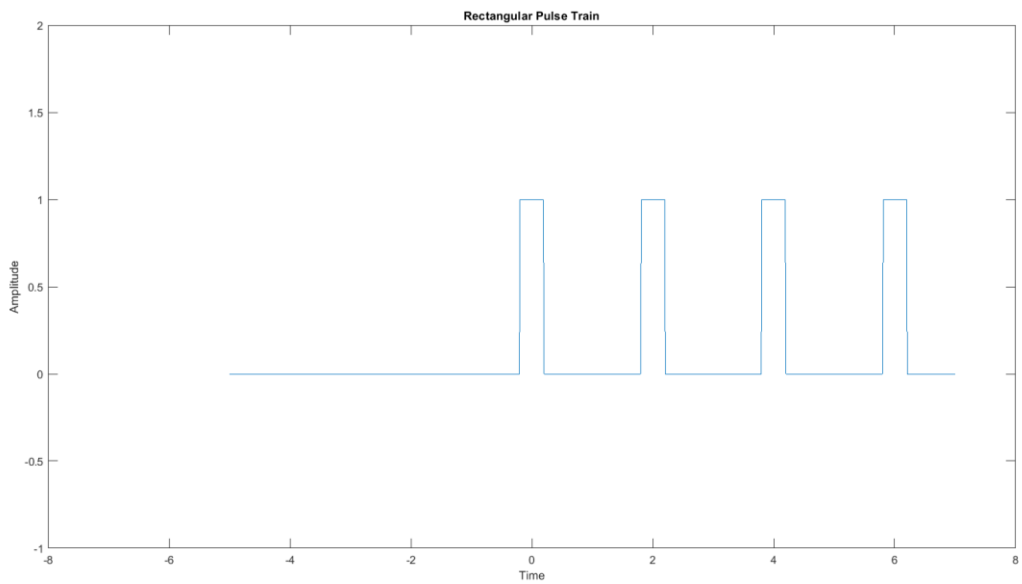
## Creating a pulse train

You can create a pulse train using the ‘pulstran’ function. It takes 3 inputs – the period of the pulse, the duration, and the type of pulse (rectangular, triangular, gaussian, etc.).

```
T = 2;                  % Period of the pulse train
pulse_duration = 0.4;   % Duration of each pulse
t = linspace(-5, 7, 1000);  % Time vector

% Generate rectangular pulse train
x = pulstran(t, 0: T: 7, 'rectpuls', pulse_duration);
```

The generated pulse is shown in Fig. 5.




*5. Fig 5*

## Code files

The associated live script file for this lab can be found here –

https://github.com/newaz-aa/Signal_Processing_Lab

*Prepared by*
*Asif Newaz*
*Lecturer, EEE, IUT*