

Date: 03/08/2025

Experiment No: 02

Experiment Name: Interpretation of basic signal in MATLAB

Theory:

MATLAB is a powerful software used in engineering and science for mathematical modeling, simulation, and signal processing. In digital signal processing (DSP), understanding basic signals is essential because they are the building blocks of more complex signals and systems.

In this lab, we use MATLAB to generate and visualize the following basic signals:

1. Unit Step Signal

- The unit step signal represents a sudden transition from 0 to 1.

Continuous-Time Definition:

A signal that is 0 for all negative time and 1 for zero and positive time.

Discrete-Time Definition:

A signal that is 0 for all negative indices and 1 from index zero onward.

2. Impulse Signal

- The impulse signal is used to analyze the response of systems.
- It has a nonzero value only at a single point in time (at zero), and zero elsewhere.

Continuous-Time Definition:

Ideally has infinite amplitude at zero and area of one. It's theoretical and not implementable in MATLAB.

Discrete-Time Definition:

A signal that is 1 at index zero and 0 elsewhere. It is practical and widely used in simulations.

3. Ramp Signal

- The ramp signal increases linearly over time and starts from zero.
- It is useful in modeling systems with steady increase or velocity.

Continuous-Time Definition:

A signal that is zero for all negative time and increases linearly for positive time.

Discrete-Time Definition:

A signal that has zero values for negative indices and increases with the index value from zero onwards.

4. Sinusoidal Signal

- The sinusoidal signal (sine or cosine) is periodic and smooth.
- It is one of the most important signals in both time and frequency domain analysis.

Continuous-Time Definition:

A signal that varies smoothly with time, described by a sine function.

Discrete-Time Definition:

A sampled version of the continuous sinusoidal signal at specific time intervals.

Codes:

```
clear all;
close all;
clc;

t = (-1:0.01:1);
d = (-1:0.1:1);

unitstep = t >= 0;
impulse = t == 0;
unitramp = t.*unitstep;

dunitstep = d >= 0;
dimpulse = d == 0;
dunitramp = d.*dunitstep;

subplot(4, 2, 1);
plot(t, unitstep, "b", "LineWidth", 1.5);
xlabel("Time (s)");
ylabel("u(t)");
title("Unit step - Continuous Time Unit")

subplot(4, 2, 3);
plot(t, impulse, "b", "LineWidth", 1.5);
xlabel("Time (s)");
ylabel("i(t)");
title("Impulse - Continuous Time Unit")

subplot(4, 2, 5);
plot(t, unitramp, "b", "LineWidth", 1.5);
xlabel("Time (s)");
ylabel("r(t)");
title("Ramp - Continuous Time Unit");

f = 50;
```

```

t_c = 0:0.000001:0.01;
y_c = 20*sin(30*pi*f*t_c);

subplot(4, 2, 7);
plot(t_c, y_c, "b", "LineWidth", 1.5);
xlabel("Time (s)");
ylabel("f(t)");
title("Continuous Sig. f(t) = (20sin(30*pi*f*t) - Continuous Time Unit");

```

```

subplot(4, 2, 2);
stem(d, dunitstep, "g", "filled", "MarkerSize", 5);
xlabel("Time (s)");
ylabel("u[t]");
title("Unit step - Discrete Time Unit")

```

```

subplot(4, 2, 4);
stem(d, dimpulse, "g", "filled", "MarkerSize", 5);
xlabel("Time (s)");
ylabel("i[t]");
title("Impluse - Discrete Time Unit")

```

```

subplot(4, 2, 6);
stem(d, dunitramp, "g", "filled", "MarkerSize", 5);
xlabel("Time (s)");
ylabel("r[t]");
title("Ramp - Discrete Time Unit")

```

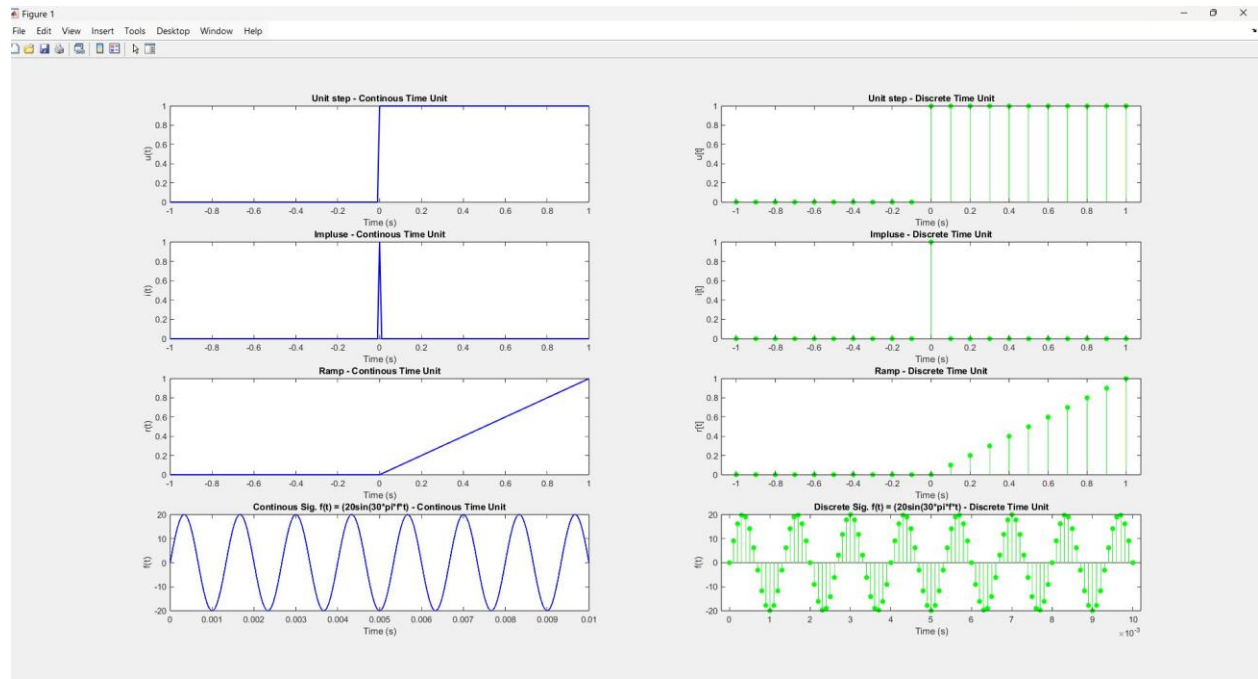
```

t_d = 0:0.0001:0.01;
y_d = 20*sin(30*pi*f*t_d);

subplot(4, 2, 8);
stem(t_d, y_d, "g", "filled", "MarkerSize", 5);
xlabel("Time (s)");
ylabel("f(t)");
title("Discrete Sig. f(t) = (20sin(30*pi*f*t) - Discrete Time Unit");

```

Output:



Discussion:

This lab helped us understand how to create and visualize basic signals like unit step, impulse, ramp, and sinusoidal in MATLAB. We saw the difference between continuous and discrete signals and how MATLAB can represent them using plots. It gave us a clear idea of how these signals behave and are used in signal processing.

Conclusion:

This lab provided a clear understanding of basic signals in both continuous and discrete forms using MATLAB. It built a strong foundation for analyzing and simulating digital signals in future DSP applications.