Aim ↬ To perform operations on signals such as addition, multiplication, scaling, shifting, folding, computation of energy & average power using MATLAB.

Software Required ↬ MATLAB

Theory ↬

Operations on signals, including addition, multiplication, scaling, shifting, and folding, allow us to modify and analyze these signals to extract meaningful insights or achieve specific objectives. These fundamental operations are essential for manipulating signals in signal processing, enabling tasks such as filtering, signal enhancement, and system analysis.

1. Addition of Signals ↪

Signal addition is the process of combining two or more signals by adding their corresponding values at each point in time. The resulting signal at any point is the sum of the values of the individual signals at that point. If x(t) and y(t) are two continuous-time signals, their addition z(t) is given by:

$$z(t) = x(t) + y(t)$$

For discrete-time signals x[n] and y[n]:

$$z[n] = x[n] + y[n]$$

2. Multiplication of Signals ↪

Signal multiplication is the process of generating a new signal by multiplying the corresponding values of two signals at each point in time. The product signal at any point is the product of the values of the individual signals at that point. For continuous-time signals x(t) and y(t), the product z(t) is:

$$z(t) = x(t) \cdot y(t)$$

For discrete-time signals x[n] and y[n]:

$$z[n] = x[n] \cdot y[n]$$

3. Time Scaling of Signals ↪

Time scaling involves compressing or expanding a signal in time. This operation changes the time duration of the signal without affecting its amplitude. For a continuous-time signal x(t) and a scalar a:

$$y(t) = x(\propto t)$$

For a discrete-time signal x[n] and a scalar a:

$$y[n] = x[\propto n]$$

If $|\propto|>1$, the signal is compressed in time (appears faster).

If $|\propto|<1$, the signal is expanded in time (appears slower).

4. Amplitude Scaling of Signals ↪

Amplitude Scaling a signal involves multiplying every value of the signal by a constant factor. This operation changes the amplitude of the signal without altering its shape. For a continuous-time signal x(t) and a scalar a:

$$y(t) = a \cdot x(t)$$

For a discrete-time signal x[n] and a scalar a:

$$y[n] = a \cdot x[n]$$

5. Shifting of Signals ↪

Shifting a signal involves moving it forward or backward in time. A right shift (or delay) means the signal occurs later in time, while a left shift (or advance) means the signal occurs earlier. For a continuous-time signal x(t):

- Right shift [delay] by T: $y(t) = x(t - T)$
- Left shift [advance] by T: $y(t) = x(t + T)$

For a discrete-time signal x[n]:

- Right shift [delay] by k: $y[n] = x[n - k]$
- Left shift [advance] by k: $y[n] = x[n + k]$

6. Time Reversal [Folding] of Signals ↪

Folding, or time reversal, of a signal is the operation of flipping the signal around the vertical axis, effectively reversing the direction of time for the signal. For a continuous-time signal x(t):

$$y(t) = x(-t)$$

For a discrete-time signal x[n]:

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

7. Computation of Energy ↪

Energy of a signal is a measure of the total magnitude of the signal over all time. For a discrete signal, it is the sum of the squares of its samples; for a continuous signal, it is the integral of the square of its magnitude over all time. Energy of a continuous-time signal x(t):

$$E = \int_{-\infty}^{\infty} |x(t)|^2 \, dt$$

For a discrete-time signal x[n]:

$$E = \sum_{n=-\infty}^{\infty} |x[n]|^2$$

8. Computation of Average Power ↪

Average Power of a signal is the average amount of energy transmitted per unit time. For a periodic signal, it is the energy per period divided by the period length; for non-periodic signals, it is typically computed over a finite duration and then averaged. Average Power of a continuous-time signal x(t) over a period T:

$$P = \frac{1}{T} \int_{T} |x(t)|^2 \, dt$$

For a discrete-time signal x[n] over N samples:

$$P = \frac{1}{N} \sum_{n=0}^{N-1} |x[n]|^2$$

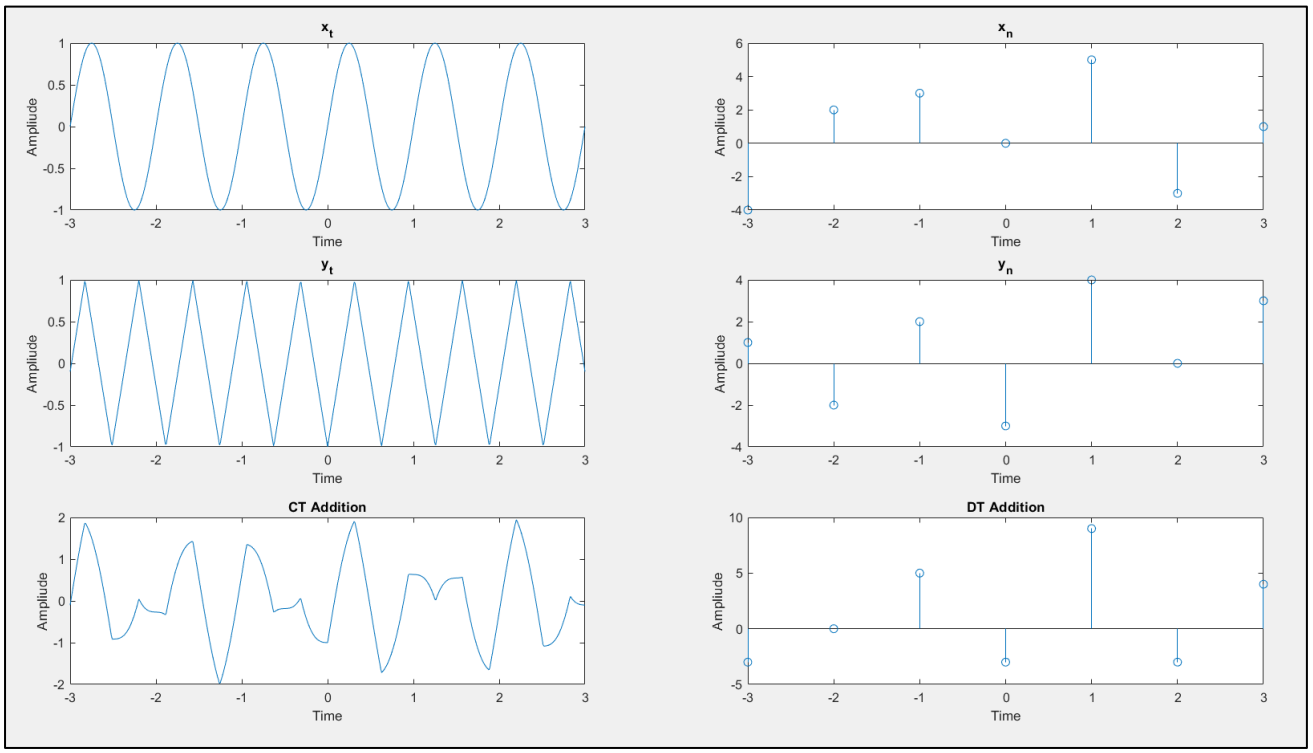
Code ↬

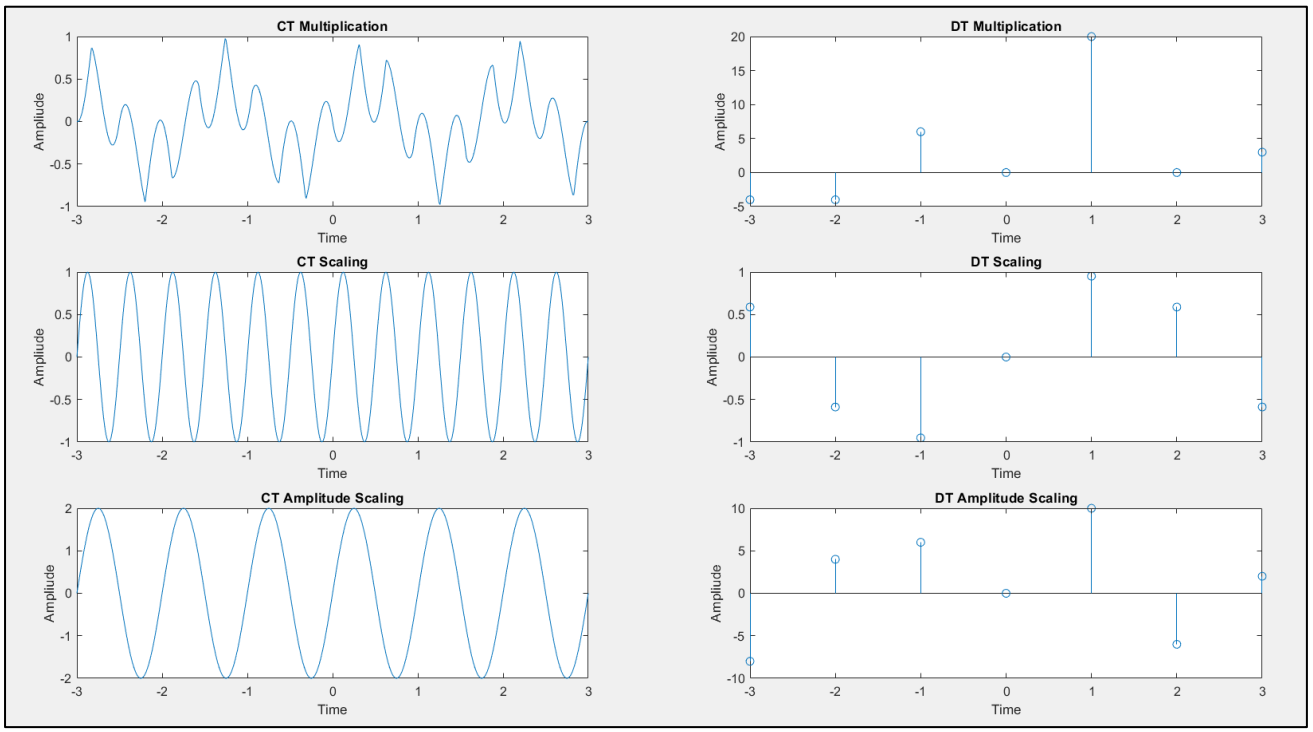
```
%Operations on Signals
clc;
clear;
t = -3:0.01:3;
n = -3:3;

%Continuous-time signal
```

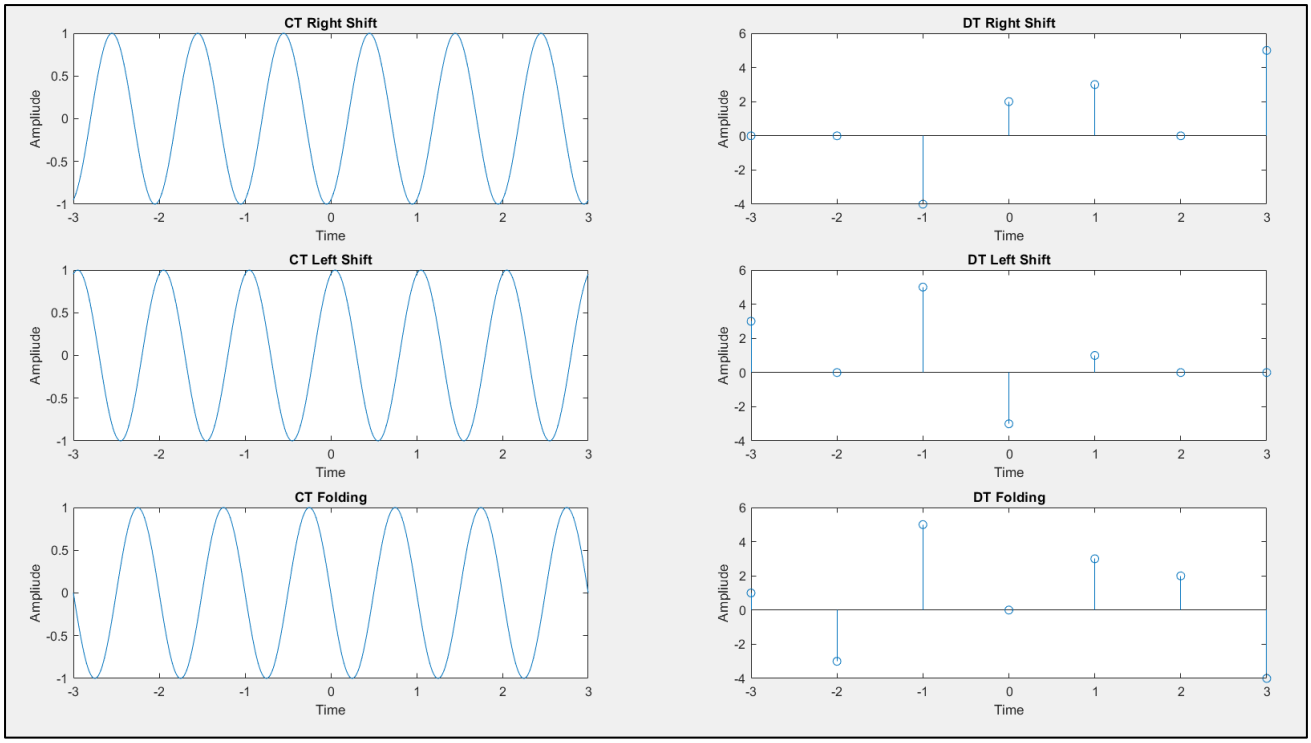
```
x_t = sin(2*pi*t);
y_t = sawtooth(10*t, 0.5);

%Discrete-time signal
x_n = [-4,2,3,0,5,-3,1];
y_n = [1,-2,2,-3,4,0,3];

%Addition
a_t = x_t + y_t;
a_n = x_n + y_n;

figure;
subplot(3,2,1); plot(t, x_t); title('x_t'); xlabel('Time'); ylabel('Ampliude');
subplot(3,2,2); stem(n, x_n); title('x_n'); xlabel('Time'); ylabel('Ampliude');
subplot(3,2,3); plot(t, y_t); title('y_t'); xlabel('Time'); ylabel('Ampliude');
subplot(3,2,4); stem(n, y_n); title('y_n'); xlabel('Time'); ylabel('Ampliude');
subplot(3,2,5); plot(t, a_t); title('CT Addition'); xlabel('Time'); ylabel('Ampliude');
subplot(3,2,6); stem(n, a_n); title('DT Addition'); xlabel('Time'); ylabel('Ampliude');

%Multiplication
m_t = x_t .* y_t;
m_n = x_n .* y_n;

%Time Scaling
a = 2;
ts_t = sin(2*pi*(a*t));
ts_n = sin(0.2*pi*(a*n));

%Amplitude Scaling
b = 2;
as_t = b * x_t;
as_n = b * x_n;

figure;
subplot(3,2,1); plot(t, m_t); title('CT Multiplication'); xlabel('Time'); ylabel('Ampliude');
subplot(3,2,2); stem(n, m_n); title('DT Multiplication'); xlabel('Time'); ylabel('Ampliude');
subplot(3,2,3); plot(t, ts_t); title('CT Scaling'); xlabel('Time'); ylabel('Ampliude');
subplot(3,2,4); stem(n, ts_n); title('DT Scaling'); xlabel('Time'); ylabel('Ampliude');
subplot(3,2,5); plot(t, as_t); title('CT Amplitude Scaling'); xlabel('Time');
ylabel('Ampliude');
```

```
subplot(3,2,6); stem(n, as_n); title('DT Amplitude Scaling'); xlabel('Time');
ylabel('Ampliude');

%Time Shifting
T = 1.2;
rs_t = sin(2*pi*(t - T));
ls_t = sin(2*pi*(t + T));

k = 2;
rs_n = [zeros(1, k), x_n(1:end-k)];
ls_n = [x_n(k+1:end), zeros(1, k)];

%Time Reversal
r_t = x_t(end:-1:1);
r_n = x_n(end:-1:1);

figure;
subplot(3,2,1); plot(t, rs_t); title('CT Right Shift'); xlabel('Time'); ylabel('Ampliude');
subplot(3,2,2); stem(n, rs_n); title('DT Right Shift'); xlabel('Time'); ylabel('Ampliude');
subplot(3,2,3); plot(t, ls_t); title('CT Left Shift'); xlabel('Time'); ylabel('Ampliude');
subplot(3,2,4); stem(n, ls_n); title('DT Left Shift'); xlabel('Time'); ylabel('Ampliude');
subplot(3,2,5); plot(t, r_t); title('CT Folding'); xlabel('Time'); ylabel('Ampliude');
subplot(3,2,6); stem(n, r_n); title('DT Folding'); xlabel('Time'); ylabel('Ampliude');

%Computation of Energy
E_t = trapz(t, abs(x_t).^2);
E_n = sum(abs(x_n).^2);

%Computation of Average Power
P_t = (1/(t(end) - t(1))) * trapz(t, abs(x_t).^2);
P_n = (1/length(x_n)) * sum(abs(x_n).^2);

% Display energy and average power
disp(['CT Signal Energy: ', num2str(E_t)]);
disp(['DT Signal Energy: ', num2str(E_n)]);
disp(['CT Signal Average Power: ', num2str(P_t)]);
disp(['DT Signal Average Power: ', num2str(P_n)]);
```

Output ↬


x_t
x_n
y_t
y_n
CT Addition
DT Addition
Ampliude
Time


CT Multiplication
DT Multiplication
CT Scaling
DT Scaling
CT Amplitude Scaling
DT Amplitude Scaling
Ampliude
Time




```
Command Window | Workspace

CT Signal Energy: 3
DT Signal Energy: 64
CT Signal Average Power: 0.5
DT Signal Average Power: 9.1429
fx >>
```

Result ↬

By using MATLAB, we successfully performed various operations on signals, including addition, multiplication, time scaling, amplitude scaling, shifting, folding, and the computation of energy and average power. The output signals were visualized and their properties analyzed.

Conclusion ↬

The MATLAB-based analysis demonstrated effective techniques for manipulating and analyzing signals. These operations are fundamental in signal processing, allowing us to understand and modify signal characteristics for various applications.

Precautions ↬

- Ensure correct sampling rates to avoid aliasing.

- Use appropriate time and amplitude scaling factors to prevent distortion.
- Verify signal lengths and indexing when performing operations on discrete-time signals.
- Handle large datasets carefully to manage computational resources efficiently.