**Aim ↬ To understand and implement the Discrete-Time Fourier Transform (DTFT) and its inverse for analyzing and synthesizing discrete-time signals using MATLAB.**

**Software Required ↬** MATLAB

**Theory ↬**

The Discrete-Time Fourier Transform (DTFT) is used to analyze discrete-time signals in the frequency domain. It transforms a discrete-time signal x[n] into a continuous function X(ω) in the frequency domain, providing insights into the signal's frequency content.

The DTFT of a discrete-time signal x[n] is defined as:

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

where:

- X(ω) is the frequency-domain representation of x[n].
- ω is the normalized angular frequency, related to the frequency f (in Hz) by ω=2πfT, with T being the sampling period.

**Properties of DTFT ↴**

1] Linearity ↪

$$F\{ax_1[n] + bx_2[n]\} = aX_1(\omega) + bX_2(\omega)$$

2] Time Shifting ↪

$$F\{x[n - n_0]\} = X(\omega)e^{-j\omega n_0}$$

3] Frequency Shifting ↪

$$F\left\{x[n]e^{j\omega_0 n}\right\} = X(\omega - \omega_0)$$

4] Convolution ↪

$$F\{x[n] * h[n]\} = X(\omega) \cdot H(\omega)$$

5] Multiplication ↪

$$F\{x[n].h[n]\} = \frac{1}{2\pi}[X(\omega) * H(\omega)]$$

6] Duality ↪

$$F\{x[n]\} = X(\omega) \quad \text{implies} \qquad F\{X[n]\} = 2\pi x(-\omega)$$

7] Time Scaling ↪

$$F\{x[an]\} = \frac{1}{|a|} X\left(\frac{\omega}{a}\right)$$

8] Differentiation in Freq Domain ↪

$$F\{nx[n]\} = j\frac{dX(\omega)}{d\omega}$$

9] Summation in Time Domain ↪

$$F\left\{\sum_{k=-\infty}^{n} x[k]\right\} = \frac{X(\omega)}{1-e^{-j\omega}}$$

10] Parseval's Energy Theorem ↪

$$E = \sum_{n=-\infty}^{\infty} |x[n]|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |X(\omega)|^2 d\omega$$

**DTFT of Basic Signals ↴**

| **x[n]** | **X(z)** |
|---|---|
| $\delta[n]$ | 1 |
| u[n] | $\frac{1}{1-e^{-j\omega}}$ |
| $a^n u[n]$ | $\frac{1}{1-ae^{-j\omega}}$ |
| $e^{j\omega_0 n}$ | $2\pi\delta(\omega-\omega_0)$ |
| rect[n] | $\frac{\sin\left(\frac{\omega N}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}$ |

| | |
|---|---|
| $\cos(\omega_0 n)$ | $\frac{\pi}{2}[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)]$ |
| $\sin(\omega_0 n)$ | $\frac{j\pi}{2}[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)]$ |

### Inverse Fourier Transform ↴

The original discrete-time signal can be reconstructed from its frequency-domain representation using the inverse DTFT:

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

This inverse transformation synthesizes the original signal from its frequency components, summing over all possible frequencies.

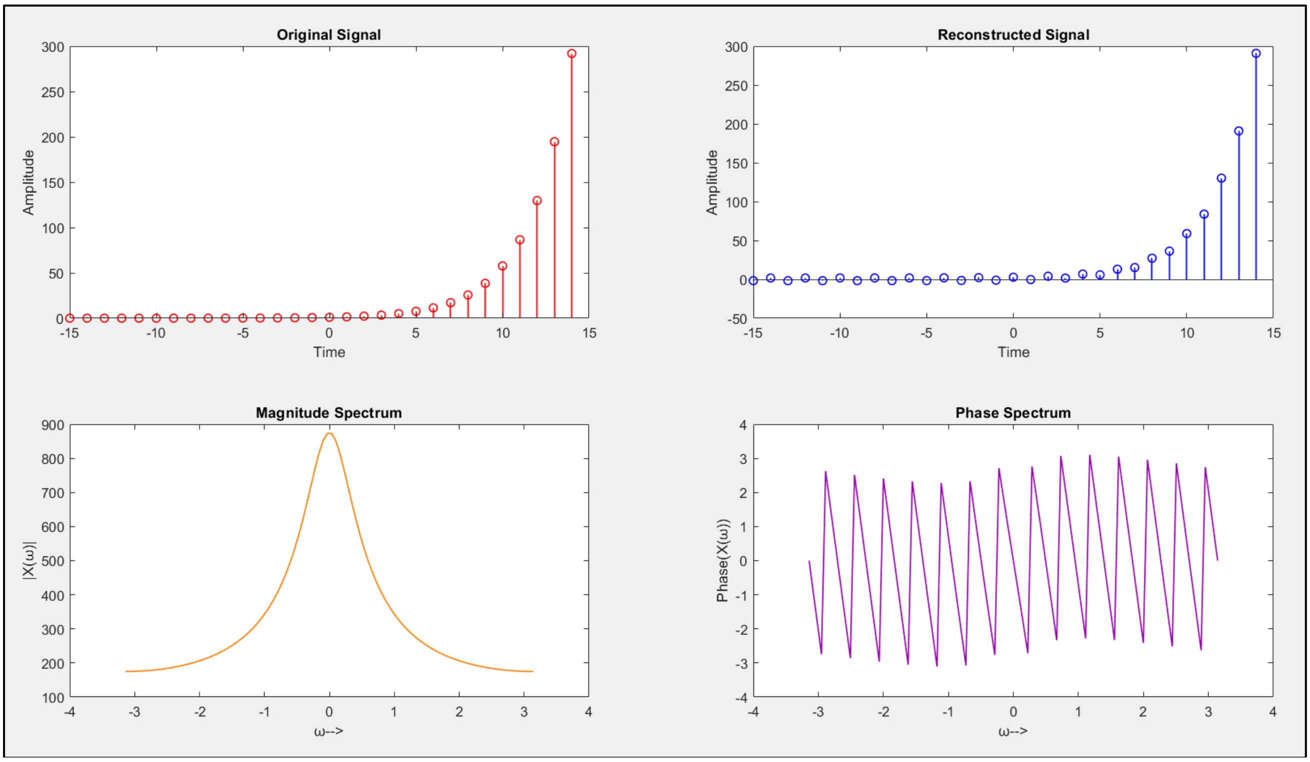
The DTFT is crucial for analyzing and processing discrete-time signals in various engineering applications, including digital signal processing and communication systems. Unlike the Discrete Fourier Transform (DFT), which works with finite sequences, the DTFT is applied to signals of infinite length, making it particularly valuable for theoretical analysis.

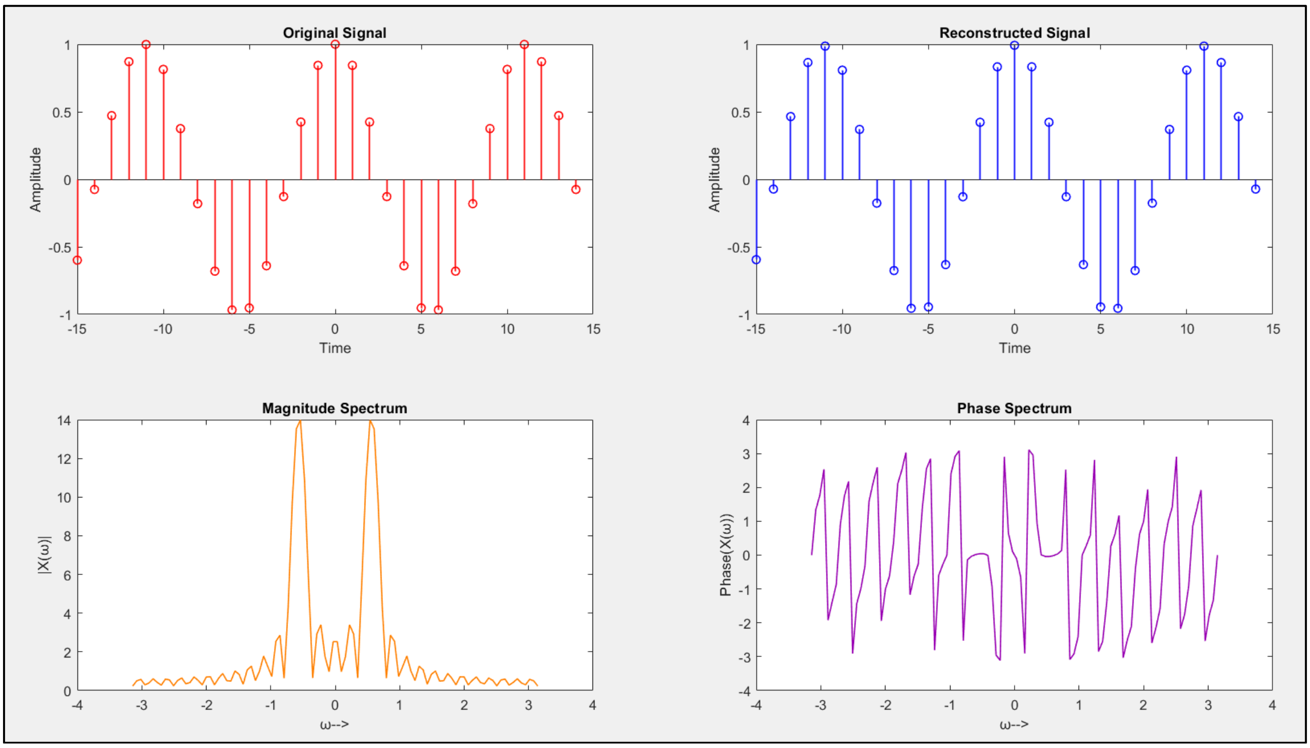
## Code ↬

```
%Discrete Time Fourier Transform
syms n a;
f_exp = a^n;
f_cos = cos(a*n);

signals = {f_exp, 1.5;
           f_cos, 12};

for i = 1:size(signals, 1)
    f = signals{i, 1};
    para = signals{i, 2};

    n0 = -15:14;
    w = linspace(-pi, pi, 100);

    f0 = double(subs(f, {a, n}, {para, n0}));
```

```
    Xw = zeros(1, length(w));
    for k = 1:length(w)
        Xw(k) = sum(f0 .* exp(-1i * w(k) * n0));
    end

    mag = abs(Xw);
    ph = angle(Xw);

    f_remake = zeros(1, length(n0));
    for k = 1:length(n0)
        f_remake(k) = (1 / length(w)) * sum(Xw .* exp(1i * w * n0(k)));
    end

    figure;
    subplot(2,2,1);
    stem(n0, f0, 'r', 'LineWidth', 1);
    xlabel('Time');
    ylabel('Amplitude');
    title('Original Signal');

    subplot(2,2,2);
    stem(n0, real(f_remake), 'b', 'LineWidth', 1);
    xlabel('Time');
    ylabel('Amplitude');
    title('Reconstructed Signal');

    subplot(2,2,3);
    plot(w, mag, 'Color', [1, 0.5, 0], 'LineWidth', 1);
    xlabel('ω-->');
    ylabel('|X(ω)|');
    title('Magnitude Spectrum');

    subplot(2,2,4);
    plot(w, ph, 'Color', [0.6, 0, 0.7], 'LineWidth', 1);
    xlabel('ω-->');
    ylabel('Phase(X(ω))');
    title('Phase Spectrum');
end
```

# Output ↬


Original Signal
Reconstructed Signal
Magnitude Spectrum
Phase Spectrum
Amplitude
Time
|X(ω)|
Phase(X(ω))
ω-->



Original Signal
Reconstructed Signal
Magnitude Spectrum
Phase Spectrum
Amplitude
Time
|X(ω)|
Phase(X(ω))
ω-->

## Result ↬

The DTFT and its inverse were effectively applied, yielding accurate frequency-domain representations and correct reconstructions of discrete-time signals. The MATLAB implementation demonstrated the DTFT's effectiveness in analyzing the frequency content of signals.

## Conclusion ↬

The DTFT is essential for analyzing discrete-time signals in the frequency domain. It provides insights into the frequency components of signals, aiding in the assessment of signal characteristics and system behavior.

## Precautions ↬

- Carefully ensure proper frequency resolution to capture significant signal details.
- Apply frequency and time-shifting properties accurately.
- Verify results by checking the inverse DTFT for accurate signal reconstruction.