Aim ↬ To understand and implement the Laplace Transform and its inverse for analyzing and synthesizing continuous-time signals using MATLAB.

Software Required ↬ MATLAB

Theory ↬

The Laplace Transform is a fundamental tool in signal processing and control systems, analogous to the Continuous-Time Fourier Transform (CTFT). It transforms a continuous-time signal from the time domain to the complex frequency domain, providing valuable insights into the system's behavior, stability, and response. The Laplace Transform is crucial for analyzing linear time-invariant (LTI) systems, solving differential equations, and designing control systems.

The Laplace Transform of a continuous-time signal x(t) is defined as:

$$X(s) = \int_0^{\infty} x(t)e^{-st}dt$$

where:

- X(s) is the Laplace Transform of x(t), representing the signal in the complex frequency domain.
- s is a complex variable, expressed as s=σ+jω, where σ is the real part (damping factor), and ω is the imaginary part (frequency).

The Laplace Transform maps a time-domain function x(t) to a function X(s) in the complex plane, revealing the poles and zeros, which are critical for understanding system stability and transient response.

Properties of Laplace Transform ↴

1] Linearity ↪

$$L\{ax_1(t) + bx_2(t)\} = aX_1(s) + bX_2(s)$$

2] Time Shifting ↪

$$L\{x(t - t_0)\} = e^{-st_0}X(s)$$

3] Frequency Shifting ↪

$$L\{e^{at}x(t)\} = X(s - a)$$

4] Time Reversal ↪

$$L\{x(-t)\} = X(-s)$$

5] Time Scaling ↪

$$L\{x(at)\} = \frac{1}{|a|} X\left(\frac{s}{a}\right)$$

6] Differentiation in Time Domain ↪

$$L\left\{\frac{d^k x(t)}{dt^k}\right\} = s^k X(s)$$

7] Integration in Time Domain ↪

$$L\left\{\int_{-\infty}^{t} x(\tau) d\tau\right\} = \frac{X(s)}{s}$$

8] Convolution in Time Domain ↪

$$L\{x(t) * y(t)\} = X(s) \cdot Y(s)$$

9] Multiplication in Time Domain ↪

$$L\{x(t). y(t)\} = \frac{1}{2\pi j} [X(s) * Y(s)]$$

10] Conjugation ↪

$$L\{x^*(t)\} = X^*(s^*)$$

11] Initial Value Theorem ↪

If X(s) is known, $\quad x(0) = \lim_{s \to \infty} X(s)$

12] Final Value Theorem ↪

If X(s) has all poles in the left half of the s-plane,

$$\lim_{t \to \infty} x(t) = \lim_{s \to 0} sX(s)$$

Laplace Transform of Basic Signals ↴

x(t)	**X(s)**
δ(t)	1
u(t)	$\frac{1}{s}$

$e^{at}u(t)$	$\frac{1}{s-a}$
$tu(t)$	$\frac{1}{s^2}$
$t^n u(t)$	$\frac{n!}{s^{n+1}}$
$\cos(\omega t)\, u(t)$	$\frac{s}{s^2+\omega^2}$
$\sin(\omega t)\, u(t)$	$\frac{\omega}{s^2+\omega^2}$

Inverse Z-Transform ↴

The Inverse Laplace Transform reconstructs the original continuous-time signal from its s-domain representation:

$$x(t) = \frac{1}{2\pi j} \oint_C X(s) e^{st} ds$$

where C is a contour in the complex plane that encircles all the poles of X(s). This integral sums the contributions of all the poles to reconstruct the time-domain signal.

The Laplace Transform and its inverse allow for the analysis and synthesis of continuous-time signals, providing a powerful framework for signal processing and system analysis in various engineering fields.

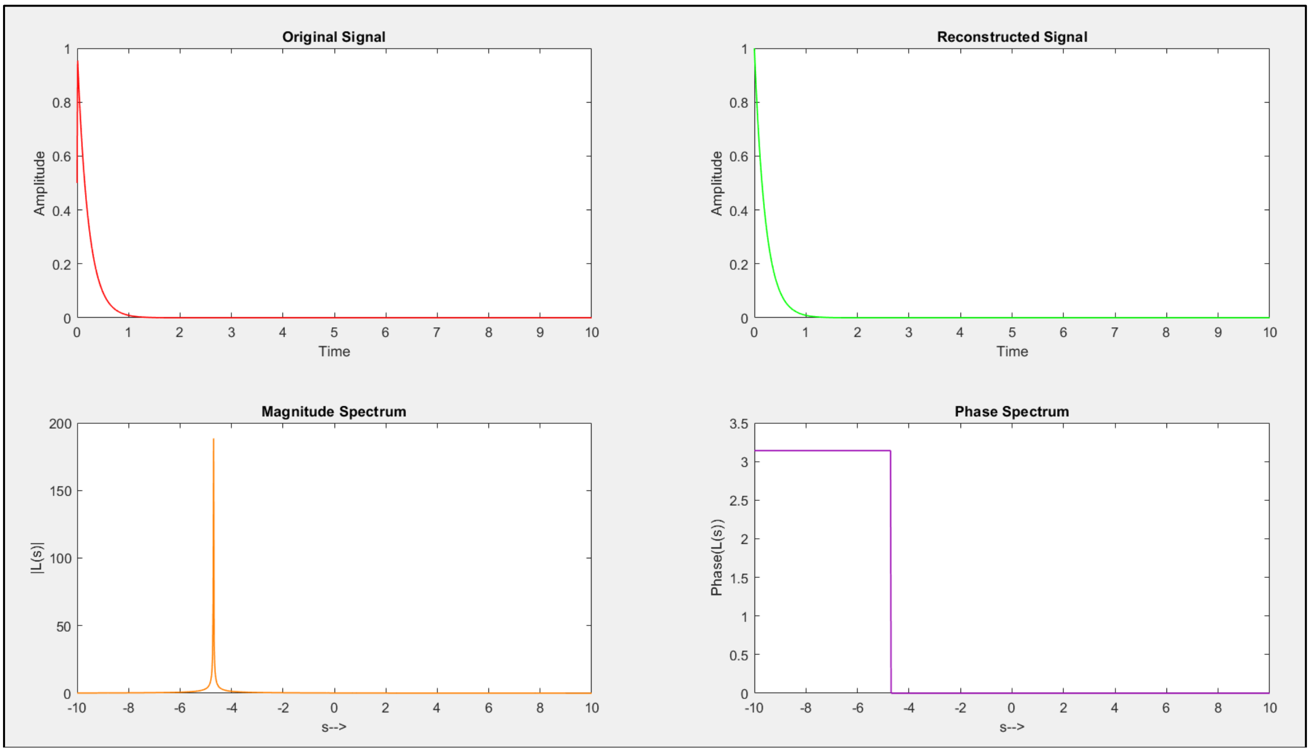
Code ↬

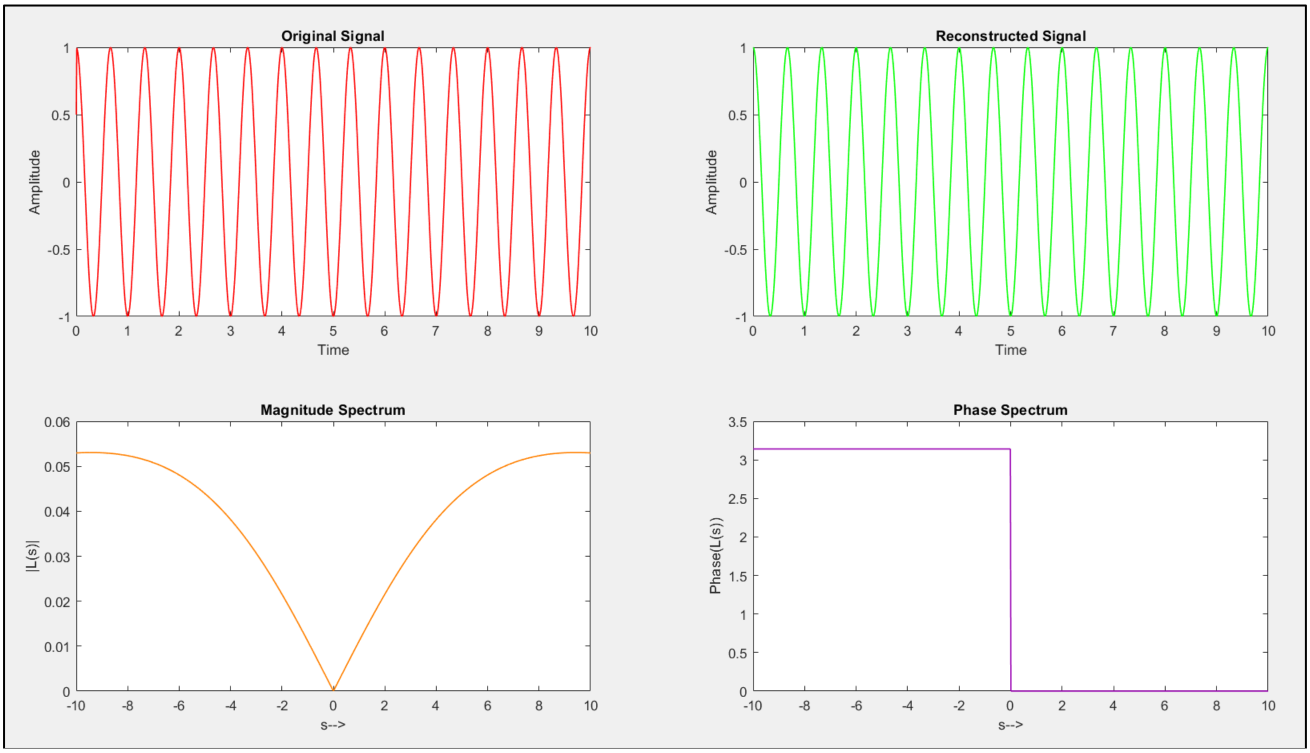
```
%Laplace Transform
syms s a t

f_exp = exp(-a * t) * heaviside(t);
f_cos = cos(a * t) * heaviside(t);

display(f_exp);
display(f_cos);

fprintf("LT of f_exp is : ");
```

```
lt_exp = laplace(f_exp);
disp(simplify(lt_exp));

fprintf("LT of f_cos is : ");
lt_cos = laplace(f_cos);
disp(simplify(lt_cos));

signals = {f_exp, 4.7;
           f_cos, 3*pi};

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

    LT = laplace(subs(f, a, para), t, s);
    ilt = ilaplace(LT);

    fprintf("ILT of %s is : ", LT);
    disp(simplify(ilt));
    LT_func = matlabFunction(LT, 'Vars', s);

    L = 1000;
    s_vals = linspace(-10, 10, L);
    t_vals = linspace(0, 10, L);

    LT_vals = LT_func(s_vals);

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

    f_numeric = double(subs(subs(f, a, para), t, t_vals));
    f_reconstruct = double(subs(subs(ilt, a, para), t, t_vals));

    figure;

    subplot(2,2,1);
    plot(t_vals, f_numeric, 'r', 'LineWidth', 1);
    xlabel('Time');
    ylabel('Amplitude');
    title('Original Signal');
```

```
    subplot(2,2,2);
    plot(t_vals, f_reconstruct, 'g', 'LineWidth', 1);
    xlabel('Time');
    ylabel('Amplitude');
    title('Reconstructed Signal');

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

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

Output ↬





Command Window | Workspace

```
>> Laplace_Transform

f_exp =

exp(-a*t)*heaviside(t)


f_cos =

cos(a*t)*heaviside(t)

LT of f_exp is : 1/(a + s)

LT of f_cos is : s/(a^2 + s^2)

ILT of 1/(s + 47/10) is : exp(-(47*t)/10)

ILT of s/(9*pi^2 + s^2) is : cos(3*pi*t)
```

Result ↬

The Laplace Transform and its inverse were effectively applied, yielding accurate s-domain representations and correct reconstructions of continuous-time signals.

The MATLAB implementation demonstrated the Laplace Transform's effectiveness in analyzing system behavior.

Conclusion ↬

The Laplace Transform is essential for analyzing continuous-time signals and systems. It simplifies differential equations into algebraic forms, aiding in the assessment of system dynamics, stability, and frequency response.

Precautions ↬

- Carefully determine the region of convergence (ROC) to ensure valid results.
- Apply properties like shifting and scaling with precision.
- Verify results by checking the inverse Laplace Transform for accuracy.