Lecture 3a: Laplace Transform with MATLAB

Dr.-Ing. Sudchai Boonto
Assistant Professor

Department of Control System and Instrumentation Engineering
King Mongkut's Unniversity of Technology Thonburi
Thailand

February 26, 2025

Laplace Transform of Simple Functions

In MATLAB by default, all functions are causal. So we don't need to multiply the functions with `heaviside(t)`. Laplace transform of simple functions using MATLAB:

```
syms t a

x1 = dirac(t);
x2 = heaviside(t);
x3 = 5*exp(a*t);

X1 = laplace(x1)
X2 = laplace(x2)
X3 = laplace(x3)
```

You will have

$$\mathcal{L}\left\{\delta(t)\right\} = 1,\ \mathcal{L}\left\{\mathbb{1}(t)\right\} = \frac{1}{s},\ \mathcal{L}\left\{5e^{at}\right\} = -\frac{5}{a-s}$$

The last one can be rewrite as $\mathcal{L}\left\{5e^{at}\right\} = \dfrac{5}{s-a}$.

Laplace Transform of Simple Functions

Laplace transform of simple functions using MATLAB:

```
syms t a omega

x1 = t;
x2 = t^2;
x3 = sin(omega*t);
x4 = cos(omega*t);

X1 = laplace(x1)
X2 = laplace(x2)
X3 = laplace(x3)
X4 = laplace(x4)
```

You will have

$$\mathcal{L}\left\{t\right\} = \frac{1}{s^2},\ \mathcal{L}\left\{t^2\right\} = \frac{2}{s^3},\ \mathcal{L}\left\{\sin(\omega t)\right\} = \frac{\omega}{\omega^2 + s^2},\ \mathcal{L}\left\{\cos(\omega t)\right\} = \frac{s}{\omega^2 + s^2}$$

Laplace Transform of Functions

We want to find the Laplace transform of the following functions:

$$f_1(t) = 1 + 5t, \quad f_2(t) = 4e^{-3t} = 10\sin(2t)$$

```
syms t a

f1 = 1 + 5*t
F1 = expand(laplace(f1))

f2 = 4*exp(-3*t) - 10*sin(2*t)
F2 = expand(laplace(f2))
```

We have

$$F1 = \frac{1}{s} + \frac{5}{s^2}, \quad F2 = \frac{4}{s+3} - \frac{20}{s^2+4}$$

Inverse Laplace Transform

We can take the inverse Laplace Transform directly without considering the partial fractions. If we have

$$F_1(s) = \frac{4}{(s+2)} + \frac{5}{(s+3)}, \quad F_2(s) = \frac{4s^2+8}{(s+2)(s+3)}$$

$$f_1(t) = 4e^{-2t} + 5e^{-3t}, \quad f_2(t) = 24e^{-2t} - 44e^{-3t} + 4\delta(t)$$

```
syms t s

F1 = 4/(s+2) + 5/(s+3)
F2 = (4*s^2 + 8)/((s+2)*(s+3))

f1 = ilaplace(F1, t)
f2 = ilaplace(F2, t)
```

Inverse Laplace Transform: Delay function

Consider

$$f(t) = \sin(2(t-2))\mathbb{1}(t-2), \quad \mathbb{1}(t) \text{ is a unit-step function.}$$
$$F(s) = e^{-2s}\frac{2}{s^2+4}$$

The MATLAB code to check the solution is

```
syms s t

f = sin(2*(t-2))*heaviside(t-2)
F = simplify(laplace(f, t, s))

ft = ilaplace(F, s, t)              % inverse back to the time-domain
```

Inverse Laplace Transform: Delay function

Obtain the inverse Laplace transform of

$$F(s) = \frac{c}{s^2}\left(1 - e^{-as}\right) - \frac{b}{s}e^{-as}, \text{ where } a > 0$$

By hand, the result is obvious as:

$$f(t) = ct - c(t-a)\mathbb{1}(t-a) - b\mathbb{1}(t-a)$$

```
syms a b c s t u(t)
assume(a > 0);              % set the condition of a

F = (c/s^2)*(1-exp(-a*s)) - (b/s)*exp(-a*s)

f = ilaplace(F, s, t);
f = subs(if1, heaviside(t-a), u(t-a))

>>f =

c*t - b*u(t - a) + c*u(t - a)*(a - t)
```

Inverse Laplace Transform

If you want to take the inverse Laplace Transform step-by-step using partial fraction, you must code more lines.

Example 1:

$$F(s) = \frac{8s+20}{(s+2)(s+3)} = \frac{4}{(s+2)} + \frac{5}{(s+3)} \quad \stackrel{\mathcal{L}}{\longleftrightarrow} \quad f(t) = 4e^{-2t} + 5e^{-3t}$$

```
syms t s

F = (8*s + 20)/((s+2)(s+3))
Fp = partfrac(F, s)
fp = children(Fp)
f1 = ilaplace(fp{1})
f2 = ilaplace(fp{2})
```

This approach works well if you know the number of terms in your partial fraction.

Inverse Laplace Transform

If you don't know the number of terms in your partial fraction, you need the for loop.

Example 2:

$$F(s) = \frac{4s^2 + 8}{(s+2)(s+3)} = 4 + \frac{24}{(s+2)} - \frac{44}{(s+3)} \xleftrightarrow{\mathcal{L}} 4\delta(t) + 24e^{-2t} - 44e^{-3t}$$

```
syms t s

F = (4*s^2 + 8)/((s+2)*(s+3))
Fp = partfrac(F, s)
Fpp = children(Fp)

n = size(Fpp,2)
for i=1:n
    vFn = sprintf('F%d', i);
    eval([vFn ' = Fpp{i}'])
    ft{i} = ilaplace(Fpp{i});
    vfn = sprintf('f%d', i);
    eval([vfn ' = ft{i}'])
end
```

Inverse Laplace Transform

In case of a system equation such as

$$\ddot{y} - 3\dot{y} + 2y = 4e^{2t}, \quad y(0^-) = -3, \text{ and } \dot{y}(0^-) = 5$$

We don't need to use the Laplace Transform to solve this ODE because MATLAB can solve it directly. However, if you want to check your understanding of how to use the Laplace Transform to solve the ODE, you can use the following codes.

```
syms y(t) t s Y

Dy = diff(y, 1); D2y = diff(y,2);
sys = D2y - 3*Dy + 2*y == 4*exp(2*t);

Fs = laplace(sys)
Fs = subs(Fs, [y(0) Dy(0)], [-3 5]; % substitute the initial conditions
Fs = subs(Fs, laplace(y(t), t, s), t, s), Y) % change to the simple version
Ys = solve(Fs, Y)                          % find the explicit solution of Y
y = ilaplace(Ys, s, t)
```

The solution is $y(t) = 4e^{2t} - 7e^{t} + 4te^{2t}$.

Inverse Laplace Transform

From previous solution, there are several ways to plot the symbolic function. The plot can be controlled by transform the symbolic function into MATLAB function by using `matlabFunction` command.

```
ts = 0:0.01:5
yt = matlabFunction(y)
plot(ts, yt(ts)); grid
```

Reference

1. Siebert, W. M., *Circuits, Signals, and Systems*, MIT Press, 1986
2. Lecture note on *Signals and Systems* Boyd, S., Stanford, USA.
3. Lathi, B. P., *Signal Processing & Linear Systems*, Berkeley-Cambridge Press, 1998
4. Hassan, K. Khalil, *Control Systems: An Introduction*, Michigan Publishing, 2023