XMUT203 Analogue Circuits and Systems

Demo 5: Laplace Transforms and Their Circuit Applications

Introduction

MATLAB has a powerful toolbox for symbolic computation. Symbolic computation enables describes the capacity to deal directly with mathematical expressions and lets us do things like differentiate or Laplace transform general expressions.

This demo merely serves as a quick introduction to the most pertinent capabilities for us. It is by no means intended as an exhaustive exploration of the symbolic toolbox. Work through the examples and feel free to explore further options and commands by consulting MATLAB help.

1. Solving Differential Equations with Calculus

In this section, we will investigate the use of several functions in symbolic toolbox of MATLAB that help us in solving differential equations of a given circuit through calculus method before we apply Laplace transform in the subsequent sections.

1.1. Solving Differential Equations

For any examples of simple differential equations, if you have a suitably defined symbol, then you can perform symbolic differentiation and integration in MATLAB. Try the following to see how this works:

```
>> clear

>> syms a x

>> diff(a*x^2)

>> int(a*x^2)
```

Furthermore, MATLAB can solve a symbolic equation and hence differential equation for you. Notice that for solving equations, you need to use == to express the “equals” sign in such expressions. If you omit the equals part, then MATLAB will assume that you wish to solve for the specified expression being zero. Sadly, it is not always able to solve complicated problems (like cubic polynomial problems) as shown by this example.

```
>> solve(x^2 + 4*x + 20)

>> solve(x^3 + 4*x + 20)
```

```
>> solve(x^3 + 6*x^2 + 11*x + 6 == 0)
```

Both the `solve` command and the related `factor` command are useful for when dealing with identifying poles of a transfer function of higher-order polynomials in preparation for partial fractions expansion.

```
>> factor(x^3 + 2*x + 20)
```

The symbolic toolbox can also deal with the solution of systems of equations:

```
>> [x,y]=solve (x + y == 8, x - y == 2)
```

The following table shows the differentiation and integration of common functions in Mathematics.

Differentiation	Integration
• $\frac{d}{dx}x^n = nx^{n-1}$ • $\frac{d}{dx}(fg) = fg' + gf'$ • $\frac{d}{dx}\left(\frac{f}{g}\right) = \frac{gf' - fg'}{g^2}$ • $\frac{d}{dx}f(g(x)) = f'(g(x))g'(x$ • $\frac{d}{dx}(\sin\ x) = \cos\ x$ • $\frac{d}{dx}(\cos\ x) = -\sin\ x$ • $\frac{d}{dx}(\tan\ x) = \sec^2\ x$ • $\frac{d}{dx}(\cot\ x) = -\csc^2\ x$ • $\frac{d}{dx}(\sec\ x) = \sec\ x \tan\ x$ • $\frac{d}{dx}(\csc\ x) = -\csc\ x \cot\ x$ • $\frac{d}{dx}(e^x) = e^x$ • $\frac{d}{dx}(a^x) = a^x \ln\ a$ • $\frac{d}{dx}\ln\ x = \frac{1}{x}$ • $\frac{d}{dx}(\arcsin x) = \frac{1}{\sqrt{1-x^2}}$ • $\frac{d}{dx}(\arctan x) = \frac{1}{1+x^2}$	• $\int a\ dx = ax + C$ • $\int \frac{1}{x}\ dx = \ln\lvert x\rvert + C$ • $\int e^x\ dx = e^x + C$ • $\int a^x\ dx = \frac{a^x}{\ln a} + C;\ a > 0,\ a \neq 1$ • $\int \ln x\ dx = x\ln x - x + C$ • $\int \sin\ x\ dx = -\cos\ x + C$ • $\int \cos\ x\ dx = \sin\ x + C$ • $\int \tan\ dx = \ln\lvert\sec\ x\rvert + C\ or\ -\ln\lvert\cos\ x\rvert +$ • $\int \cot\ x\ dx = \ln\lvert\sin\ x\rvert + C$ • $\int \sec\ x\ dx = \ln\lvert\sec\ x + \tan\ x\rvert + C$ • $\int \csc\ x\ dx = \ln\lvert\csc\ x - \cot\ x\rvert + C$ • $\int \sec^2\ x\ dx = \tan\ x + C$ • $\int \sec\ x\ \tan\ x\ dx = \sec\ x + C$ • $\int \csc^2\ x\ dx = -\cot\ x + C$ • $\int \tan^2\ x\ dx = \tan\ x - x + C$ • $\int \frac{dx}{\sqrt{a^2-x^2}} = \arcsin\left(\frac{x}{a}\right) + C$ • $\int \frac{dx}{\sqrt{a^2+x^2}} = \ln\left\lvert\frac{x+\sqrt{a^2+x^2}}{a}\right\rvert + C$

Table 1: Selected differentiation and integration of common functions

Exercise 1 – Solving Differential Equations

A linear circuit has an impulse response $h(t) = 2e^{-t}\ u(t)$ and an input $x(t) = e^{-2t}u(t)$. Find the steady-state response using:

a. The time-domain calculus integral.

$$y(t) = \int_0^t h(t-\tau)\,x(\tau)d\tau$$

b. The s-domain process.

$$y(t) = \mathcal{L}^{-1}\{H(s)X(s)\}$$

c. Comment on the results in parts (a) and (b).

Solution

a. Meanwhile, back in the t-domain the convolution integral yields:

$$\begin{aligned} y(t) &= \int_0^t h(t-\tau)\,x(\tau)d\tau \\ &= \int_0^t 2e^{-(t-\tau)}\,e^{-2\tau}d\tau \\ &= 2e^{-t}\int_0^t e^{\tau}\,e^{-2t}d\tau \\ &= 2e^{-t}\int_0^t e^{-\tau}\,d\tau \\ &= 2e^{-t} - 2e^{-2t} \quad \text{for} \quad t \geq 0 \end{aligned}$$

b. Converting $h(t)$ and $x(t)$ into the s-domain yields:

$$H(s) = \mathcal{L}\{2e^{-t}\} = \frac{2}{s+1}$$

And

$$X(s) = \mathcal{L}\{e^{-2t}\} = \frac{1}{s+2}$$

Applying equation below produces $y(t)$ as:

$$\begin{aligned} y(t) &= \mathcal{L}^{-1}\{H(s)X(s)\} \\ &= \mathcal{L}^{-1}\left\{\frac{2}{(s+1)(s+2)}\right\} \\ &= \mathcal{L}^{-1}\left\{\frac{2}{(s+1)} - \frac{2}{(s+2)}\right\} \\ &= 2e^{-t} - 2e^{-2t} \quad \text{for} \quad t > 0 \end{aligned}$$

c. The two methods produce the same result. The difference is that the convolution integral evaluation is carried out entirely in the time domain.

1.2. Partial Fractions Expansion and Related Matters

To help us perform the Laplace transform manually, a particularly useful MATLAB command for us to use is `partfrac`, which solves partial fractions expansion.

```
>> clear
>> syms s
>> Y = (3 * s + 13) / (s^2 + 4 * s + 3)
>> Yp = partfrac(Y)
```

The following table lists the partial fraction expansion of several selected equation forms in the circuit analysis.

#Pair	$\boldsymbol{F(s)}$	$\boldsymbol{f(t)}$	**Description**
1	$\frac{K}{s+a}$	$Ke^{-at}u(t)$	Distinct real
2	$\frac{K}{(s+a)^2}$	$Kte^{-at}u(t)$	Repeated real
3	$\frac{K}{s+\alpha-j\beta}+\frac{K^*}{s+\alpha+j\beta}$	$2\|K\|e^{-at}\cos(\beta t+\theta)\,u(t)$	Distinct complex
4	$\frac{K}{(s+\alpha-j\beta)^2}+\frac{K^*}{(s+\alpha+j\beta)^2}$	$2t\|K\|e^{-at}\cos(\beta t+\theta)\,u(t)$	Repeated complex

Note: in pairs 1 and 2, K is a real quantity, whereas in pairs 3 and 4, K is the complex quantity $|K|\angle\theta$.

Table 2: Useful partial fraction expansion in the circuit analysis

Exercise 2 - Partial Fraction Expansions

Given the following transfer function equation of a circuit.

$$F(s) = \frac{(s+1)(s+3)}{s(s+2)(s+4)}$$

Use MATLAB's partfrac() function to perform partial fraction expansion of the equation.

Solution

The following MATLAB code is used to compute the partial fraction expansion and the inverse Laplace transform.

MATLAB Code (partialfraction.m)

% partialfraction.m

```
% Declare symbol and function
syms s
F=(s+1)*(s+3)/(s*(s+2)*(s+4))

% Apply partial fractional expansion
Factorized=partfrac(F)

% Appy inverse Laplace transform
ilaplace(Factorized)
```

The following output is generated.

```
F =
((s + 1)*(s + 3))/(s*(s + 2)*(s + 4))
Factorized =
1/(4*(s + 2)) + 3/(8*(s + 4)) + 3/(8*s)
ans =
exp(-2*t)/4 + (3*exp(-4*t))/8 + 3/8
```

Notice that we used also the MATLAB function ilaplace() to compute the inverse Laplace transform.

2. Symbolic Laplace and Inverse Laplace Transforms

You can quickly find the Laplace and inverse Laplace transforms using MATLAB's `laplace` and `ilaplace` commands. Use `heaviside(t)` to define a unit step and `dirac` to specify a dirac delta function.

For example, let's apply a signal $y(t) = 3u(t-2) + \sin(\omega t)$ to a system having transfer function $G(s) = 3/(s+3)$ and find the output signal.

```
>> clear
>> syms w t s
>> X = laplace(3 * heaviside(t-2) + sin(w * t))
>> G = 3/(s+3)
>> x= ilaplace(X*G)
```

The following table outlines some selected forward Laplace transform and their relevant inverse transforms for several general-purpose functions typically found in the circuit analysis and some relevant Laplace transform's properties.

Function	$f(t)(t > 0^-)$	$F(s)$
Impulse	$\delta(t)$	1
Step	$u(t)$	$\frac{1}{s}$
Ramp	t	$\frac{1}{s^2}$
Exponential	e^{-at}	$\frac{1}{s+a}$
Sine	$\sin \omega t$	$\frac{\omega}{s^2+\omega^2}$
Cosine	$\cos \omega t$	$\frac{s}{s^2+\omega^2}$
Damped ramp	te^{-at}	$\frac{1}{(s+a)^2}$
Damped sine	$e^{-at} \sin \omega t$	$\frac{\omega}{(s+a)^2+\omega^2}$
Damped cosine	$e^{-at} \cos \omega t$	$\frac{s+a}{(s+a)^2+\omega^2}$

Table 3: Selected Laplace transforms and their inverse transforms.

$f(t)$	$F(s)$	**Operation**
$Kf(t)$	$KF(s)$	Multiplication by a constant
$f_1(t) + f_2(t) + f_3(t)$	$F_1(s) + F_2(s) + F_3(s)$	Addition/subtraction
$\frac{df(t)}{dt}$	$sF(s) - f(0^-)$	First derivative (time)
$\frac{d^2f(t)}{dt^2}$	$s^2F(s) - sf(0^-) - \frac{df(0^-)}{dt}$	Second derivative (time)
$\frac{d^nf(t)}{dt^n}$	$s^nF(s) - s^{n-1}f(0^-) - s^{n-2}\frac{df(0^-)}{dt} - s^{n-3}\frac{d^2f(0^-)}{dt^2} - \cdots - \frac{d^{n-1}f(0^-)}{dt^{n-1}}$	n-th derivative (time)

$\int_0^t f(x)\,dx$	$\frac{F(s)}{s}$	Time integral
$f(t-a)u(t-a),\ a > 0$	$e^{-as}F(s)$	Translation in time
$f(at),\quad a > 0$	$\frac{1}{a}F\left(\frac{s}{a}\right)$	Scale changing
$tf(t)$	$-\frac{dF(s)}{ds}$	First derivative (s)
$t^n f(t)$	$(-1)^n \frac{d^n F(s)}{ds^n}$	n-th derivative (s)
$\frac{f(t)}{t}$	$\int_s^\infty F(u)\,du$	s integral

Table 4: Selected properties of Laplace transform

Exercise 3 – Laplace Transform Method vs. Calculus Method

A linear circuit has an impulse response $h(t) = e^{-100t}u(t)$ and an input $x(t) = tu(t)$.

a. Find the zero-state response using the t-domain convolution integral.

b. Validate your answer using both the s-domain process.

c. Verify the result of integration and Laplace transform by simulating the circuit in MATLAB.

d. Which method was the easier to use?

Solution

a. Using the convolution integral for this example circuit yields:

$$y(t) = \int_0^t h(t-\tau)\,x(\tau)d\tau$$

$$= \int_0^t e^{-100(t-\tau)}\tau d\tau$$

$$= e^{-100t}\int_0^t e^{100t}\,\tau d\tau$$

$$= e^{-100t}\left[e^{-100\tau}\left(\frac{\tau}{100} - \frac{1}{10^4}\right)\right]_0^t$$

$$= e^{-100t}\left[e^{-100\tau}\left(\frac{\tau}{100} - \frac{1}{10^4}\right) - 1\left(\frac{0}{100} - \frac{1}{10^4}\right)\right]$$

$$= \left(\frac{t}{100}\right)u(t) - \left(\frac{1}{10^4}\right)u(t) + \left(\frac{e^{-100t}}{10^4}\right)u(t)$$

b. Using the s-domain process, we first convert $h(t)$ and $x(t)$ into the s-domain as follows:

$$H(s) = \mathcal{L}\{e^{-100t}u(t)\} = \frac{1}{s+100}$$

And

$$X(s) = \mathcal{L}\{tu(t)\} = \frac{1}{s^2}$$

The output of the linear circuit is:

$$Y(s) = H(s)X(s)$$

Entering the functions into the equation above, it becomes:

$$Y(s) = \frac{1}{s^2}\frac{1}{(s+100)}$$

$$= \frac{1}{s}\left[\frac{1}{s(s+100)}\right]$$

Applying partial fraction expansion, the equation above becomes:

$$Y(s) = \frac{1}{s}\left[\frac{0.01}{s} - \frac{0.01}{s+100}\right]$$

$$= \frac{0.01}{s^2} - \frac{0.01}{s(s+100)}$$

$$= \frac{0.01}{s^2} - \frac{10^{-4}}{s} + \frac{10^{-4}}{s+100}$$

Taking inverse Laplace transform, the equation in the time domain is:

$$y(t) = \mathcal{L}^{-1}\{Y(s)\} = \left(\frac{t}{100}\right)u(t) - \left(\frac{1}{10^4}\right)u(t) + \left(\frac{e^{-100t}}{10^4}\right)u(t)$$

c. The following MATLAB program could be used to validate the s-domain process

MATLAB Code (integrallaplace.m)

```
% clear all
% Define variables to use
syms t s

% Declare functions of the circuit
h = exp(-100*t);
x = t;

% Take Laplace transform
H = laplace(h);
X = laplace(x);
```

```
% Perform multiplication
Y = H*X;

% Take inverse Laplace transform
y = ilaplace(Y)
```

This produces the following result that is the same as that found by hand:

```
y = t/100 + exp(-100*t)/10000 - 1/10000
```

d. For most of cases, it appeared that the s-domain process was easier than t-domain process. But it is not always the case, as this example shows.

The following exercise shows the use of the ilaplace function from the symbolic toolbox of MATLAB to perform inverse Laplace transform. Notice the use of pretty function in MATLAB to better format the display of the equation in the screen.

Exercise 4 - Inverse Laplace Transform of a Function

Use MATLAB to find the inverse transform $f(t)$ of the following function:

$$F(s) = \frac{(s+100)^2}{(s+50)^2(s+200)}$$

Solution

The following MATLAB code using the `ilaplace` function will find the desired waveform:

MATLAB Code (inverselaplace.m)

```
% Declare variables to be used
syms t s

% Define the transfer function
F = (s+100)^2/(s+50)^2/(s+200);

% Retrieve inverse Laplace transform
f = ilaplace(F)
pretty (f)
```

MATLAB returns the following answer:

```
f = (5*exp(-50*t))/9 + (4*exp(-200*t))/9 + (50*t*exp(-50*t))/3
```

The pretty function helps MATLAB express the answer in a more recognizable form.

```
exp(-50 t) 5   exp(-200 t) 4   t exp(-50 t) 50
------------ + ------------- + ---------------
      9              9                3
```

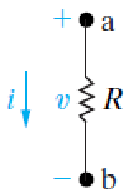
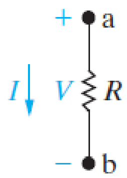
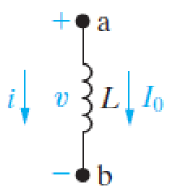
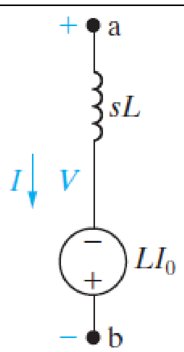
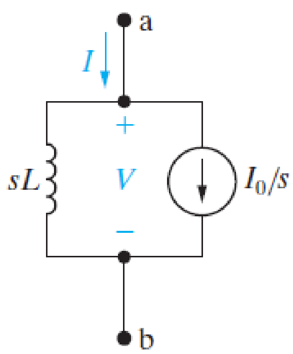
Or, written in a more common way

$$f(t) = \left(\frac{5}{9}\right)e^{-50t} + \left(\frac{4}{9}\right)e^{-200t} + \left(\frac{50t}{3}\right)e^{-50t}$$

3. Circuit Applications of Laplace Transform

When used for circuit analysis, Laplace transform is useful for simplifying the calculation processes. Instead of using complicated calculus in terms of differentiation and integration to solve the differential equations of the circuits, we could resort to simple algebra processes to solve them.

Before we can perform these processes, it is essential that we transform the components and devices in the circuit into their equivalent s-domain formats. We limit our scope to just R, L, and C devices in the circuit in this course. Some of notable transformation of these devices to their s-domain equivalents are tabulated below.

<table>
<tr><th>Time Domain</th><th colspan="2">Frequency Domain</th></tr>
<tr><td>
</td><td>
</td><td></td></tr>
<tr><td>$v = Ri$</td><td>$V = RI$</td><td></td></tr>
<tr><td>
</td><td>
</td><td>
</td></tr>
<tr><td>$v = L\left(\frac{di}{dt}\right)$</td><td>$V = sLI - LI_0$</td><td>$I = \frac{V}{sL} + \frac{I_0}{s}$</td></tr>
</table>

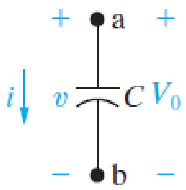
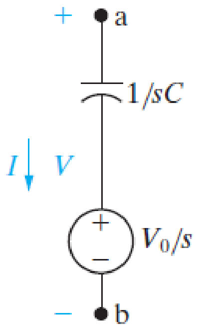
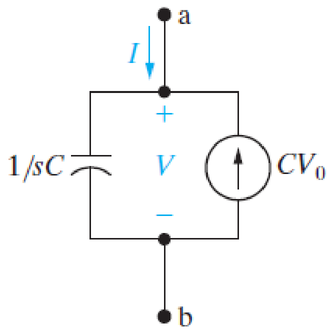
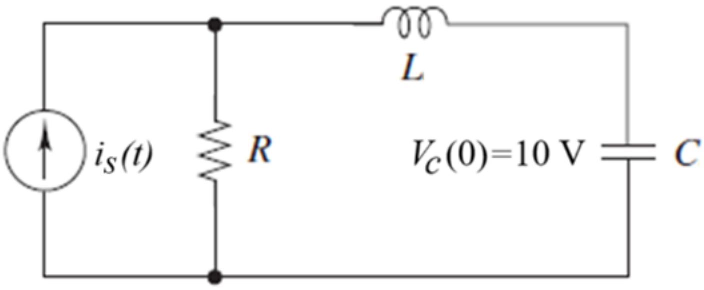
$i = \frac{1}{L}\int_0^t v\,dx + I_o$		
		
$i = C\left(\frac{dv}{dt}\right)$ $v = \frac{1}{C}\int_0^t i\,dx + V_o$	$v = \frac{1}{sC} + \frac{V_0}{s}$	$I = sCV - CV_0$

Table 5: Transformation of devices in the circuit to their s-domain equivalents

Exercise 5 – Laplace Transform in Solving Mesh Current Equation

For the schematic diagrams of a second-order series RLC circuit given below, perform the following tasks.



a. Applying Laplace transform, determine the Thevenin and Norton equivalents of the circuit.

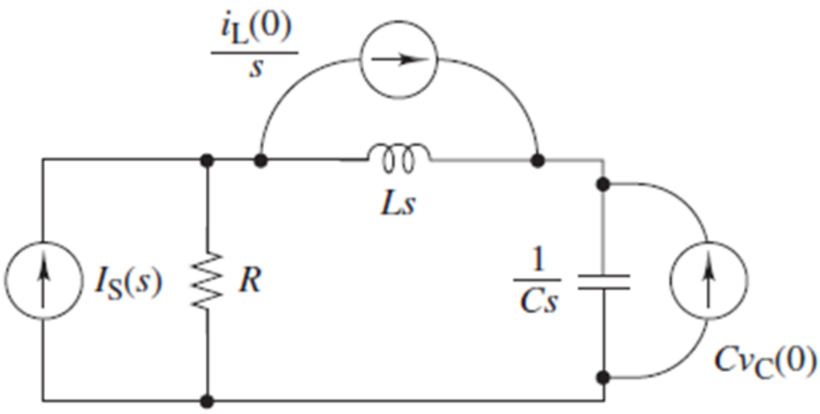
b. Formulate the s-domain mesh-current equation for the circuit.

c. Solve for the natural response of the circuit with $i_s(t) = 0$ for mesh current in the circuit as $i_L(0) = 0$, $v_C(0)$ = 10 V, L = 250 mH, C = 1 μF, and R = 1 kΩ.

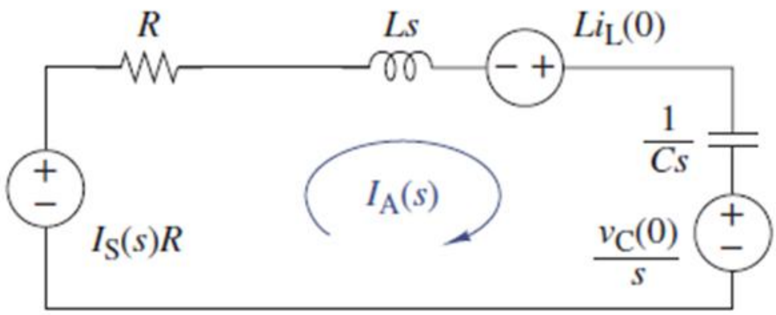
d. Use LTspice to simulate the circuit in figure (b) and then validate your results by running a MATLAB simulation of the circuit. Compare the two plots.

Solution

a. Figure below is the s-domain circuit. In this circuit, the current source is connected in parallel with the impedance.



The source transformations convert the current sources in the circuit above into their equivalent voltage sources as shown in the figure below.



b. By inspection, the KVL equation for the single mesh in this circuit is:

$$\left(R + Ls + \frac{1}{Cs}\right) I_A(s) = RI_S(s) + Li_L(0) - \frac{v_C(0)}{s}$$

With $I_S = 0$, $i_L(0) = 0$ and $v_C(0)$ = 10 V, thus:

$$I_A(s) = \frac{RI_S(s) + Li_L(0) - \frac{v_C(0)}{s}}{\left(R + Ls + \frac{1}{Cs}\right)} = \frac{RCsI_S(s) - CV_c(0)}{LCs^2 + RCs + 1}$$

The roots in the denominator in the equation above are roots of the quadratic equation $LCs^2 + RCs + 1 = 0$, which we recognize as the characteristic equation of a series RLC circuit.

In the previous topic on RLC circuits, we called these roots of the natural frequencies. Thus, the natural poles of the circuit are its natural frequencies.

c. Solving the mesh equation for the current I_A yields:

$$I_{Azi}(s) = -\left(\frac{10^{-5}}{0.25\times 10^{-6}s^2 + 10^{-3}s + 1}\right)$$
$$= \frac{-40}{s^2 + 4\times 10^3 s + 4\times 10^6}$$
$$= \frac{-40}{(s+2000)^2}$$

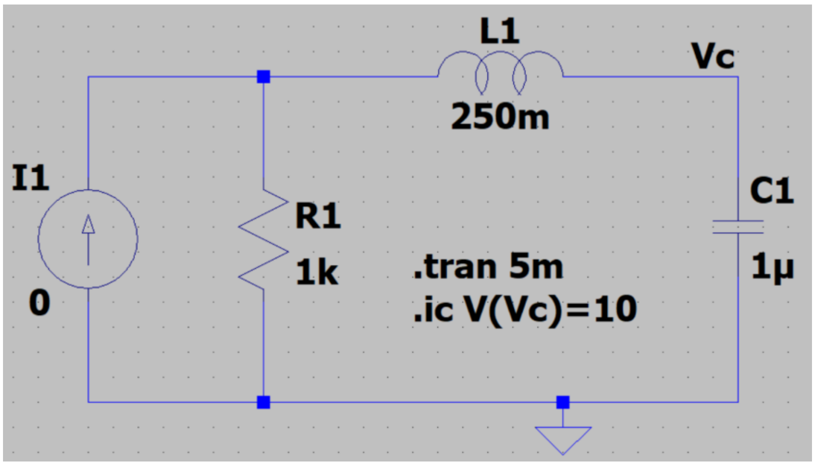
The zero-state response has two identical natural poles, both located at s = −2000. The inverse transform of $I_{Azi}(s)$ is a damped ramp waveform:

$$I_{Azi}(s) = -(40te^{-2000t})u(t)\ \mathrm{A}$$

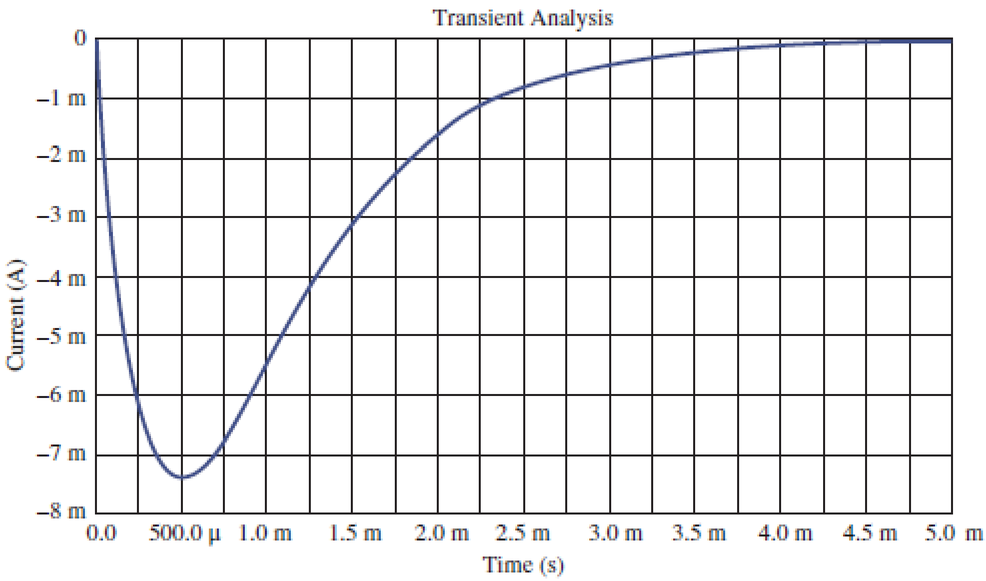
The damped ramp response indicates a critically damped second-order circuit. The minus sign means the direction of the actual current is opposite to the reference mark assigned to $I_A(s)$ in the circuit above.

This sign makes sense physically since the capacitor initial condition source in the circuit tends to drive current in a direction opposite to the assigned reference mark.

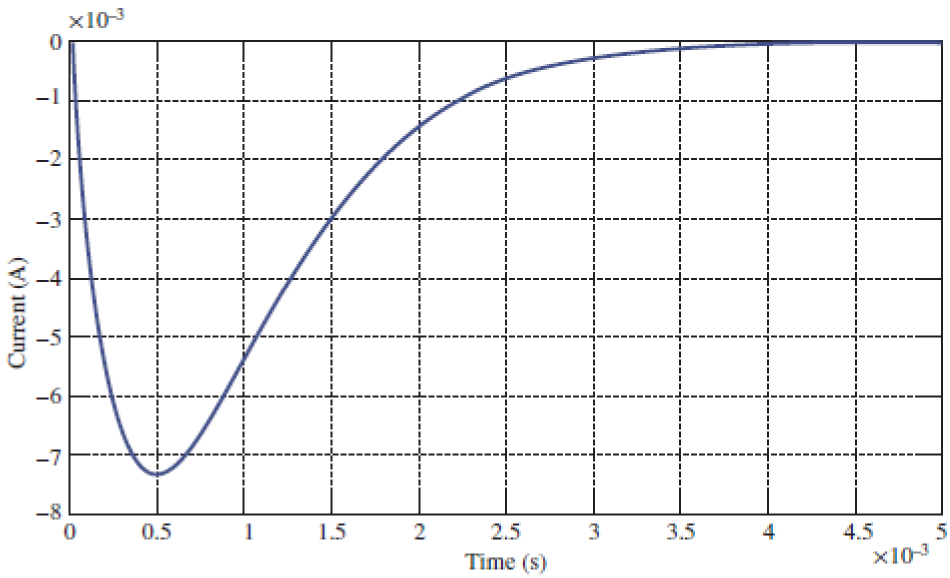
d. In the LTspice, a series RLC circuit as shown below is built with the initial condition of the capacitor set at +10 V and that of the inductor to 0 A. There was no input since the requirement was for the natural response.



A transient response of the circuit is simulated using LTspice. The transient analysis of the circuit current was plotted versus time for 5 ms. The resulting plot is shown in the figure below.



Then, MATLAB was used to plot the result of the transient response shown below.



The following MATLAB code was used to simulate the transient response.

MATLAB Code (meshlaplace.m)

```
% Specify range of time
t = 0:2e-6:5e-3;

% Declare function
iR=-40*t.*exp(-2000*t);

% Plot the graph
plot(t,iR)
grid on
xlabel('Time (s)')
ylabel('Current (A)')
```

Comparing the results shows no measurable differences between the two simulation plots. The waveform and magnitude of the transient responses with LTspice and MATLAB are found to be identical.

Appendix

`solve()`

Equations and systems solver

Syntax

S = solve(eqn,var)

S = solve(eqn,var,Name=Value)

Y = solve(eqns,vars)

Y = solve(eqns,vars,Name=Value)

[y1,...,yN] = solve(eqns,vars)

[y1,...,yN] = solve(eqns,vars,Name=Value)

[y1,...,yN,parameters,conditions] = solve(eqns,vars,ReturnConditions=true)

Description

S = solve(eqn,var) solves the symbolic equation eqn for the variable var. If you do not specify var, the symvar function determines the variable to solve for. For example, solve(x + 1 == 2, x) solves the equation $x + 1 = 2$ for x.

S = solve(eqn,var,Name=Value) uses additional options specified by one or more Name=Value arguments.

Y = solve(eqns,vars) solves the system of equations eqns for the variables vars and returns a structure that contains the solutions. If you do not specify vars, solve uses symvar to find the variables to solve for. In this case, the number of variables that symvar finds is equal to the number of equations eqns.

Y = solve(eqns,vars,Name=Value) uses additional options specified by one or more Name=Value arguments.

[y1,...,yN] = solve(eqns,vars) solves the system of equations eqns for the variables vars. The solutions are assigned to the variables y1,...,yN. If you do not specify the variables, solve uses symvar to find the variables to solve for. In this case, the number of variables that symvar finds is equal to the number of output arguments N.

[y1,...,yN] = solve(eqns,vars,Name=Value) uses additional options specified by one or more Name=Value arguments.

[y1,...,yN,parameters,conditions] = solve(eqns,vars,ReturnConditions=true) returns the additional arguments parameters and conditions that specify the parameters in the solution and the conditions on the solution.

Examples

Solve Quadratic Equation

Solve the quadratic equation without specifying a variable to solve for. solve chooses x to return the solution.

syms a b c x

eqn = a*x^2 + b*x + c == 0

eqn =

$$ax^2 + bx + c = 0$$

S = solve(eqn)

S =

$$\begin{pmatrix} \dfrac{b + \sqrt{b^2 - 4ac}}{2a} - \\ -\dfrac{b - \sqrt{b^2 - 4ac}}{2a} \end{pmatrix}$$

Specify the variable to solve for and solve the quadratic equation for a.

Sa = solve(eqn,a)

Sa =

$$-\left(\frac{c + bx}{x^2}\right)$$

`partfrac()`

Partial fraction decomposition

Syntax

partfrac(expr,var)

partfrac(expr,var,Name,Value)

Description

partfrac(expr,var) finds the partial fraction decomposition of expr with respect to var. If you do not specify var, then partfrac uses the variable determined by symvar.

partfrac(expr,var,Name,Value) finds the partial fraction decomposition using additional options specified by one or more Name,Value pair arguments.

Examples

Partial Fraction Decomposition of Symbolic Expressions

Find partial fraction decomposition of univariate and multivariate expressions.

First, find partial fraction decomposition of univariate expressions. For expressions with one variable, you can omit specifying the variable.

syms x

partfrac(x^2/(x^3 - 3*x + 2))

ans =

$$5/(9*(x - 1)) + 1/(3*(x - 1)\hat{}2) + 4/(9*(x + 2))$$

`laplace()`

Laplace transform

Syntax

F = laplace(f)

F = laplace(f,transVar)

F = laplace(f,var,transVar)

Description

F = laplace(f) returns the Laplace Transform of f. By default, the independent variable is t and the transformation variable is s.

F = laplace(f,transVar) uses the transformation variable transVar instead of s.

F = laplace(f,var,transVar) uses the independent variable var and the transformation variable transVar instead of t and s, respectively.

Examples

Laplace Transform of Symbolic Expression

Compute the Laplace transform of 1/sqrt(x). By default, the transform is in terms of s.

syms x y

f = 1/sqrt(x);

F = laplace(f)

F =

$$\frac{\sqrt{\pi}}{\sqrt{s}}$$

Specify Independent Variable and Transformation Variable

Compute the Laplace transform of exp(-a*t). By default, the independent variable is t, and the transformation variable is s.

syms a t y

f = exp(-a*t);

F = laplace(f)

F =

$$\frac{1}{a+s}$$

Specify the transformation variable as y. If you specify only one variable, that variable is the transformation variable. The independent variable is still t.

F = laplace(f,y)

F =

$$\frac{1}{a+y}$$

Specify both the independent and transformation variables as a and y in the second and third arguments, respectively.

F = laplace(f,a,y)

F =

$$\frac{1}{t+y}$$

`ilaplace()`

Inverse Laplace transform

Syntax

f = ilaplace(F)

f = ilaplace(F,transVar)

f = ilaplace(F,var,transVar)

Description

f = ilaplace(F) returns the Inverse Laplace Transform of F. By default, the independent variable is s and the transformation variable is t. If F does not contain s, ilaplace uses the function symvar.

f = ilaplace(F,transVar) uses the transformation variable transVar instead of t.

f = ilaplace(F,var,transVar) uses the independent variable var and the transformation variable transVar instead of s and t, respectively.

Examples

Inverse Laplace Transform of Symbolic Expression

Compute the inverse Laplace transform of 1/s^2. By default, the inverse transform is in terms of t.

syms s

F = 1/s^2;

f = ilaplace(F)

$$f = t$$

Default Independent Variable and Transformation Variable

Compute the inverse Laplace transform of 1/(s-a)^2. By default, the independent and transformation variables are s and t, respectively.

syms a s

F = 1/(s-a)^2;

f = ilaplace(F)

$$f = te^{at}$$

Specify the transformation variable as x. If you specify only one variable, that variable is the transformation variable. The independent variable is still s.

syms x

f = ilaplace(F,x)

$$f = xe^{ax}$$

Specify both the independent and transformation variables as a and x in the second and third arguments, respectively.

f = ilaplace(F,a,x)

$$f = xe^{sx}$$