Laboratory 4 Symbolic Maths & Circuits 1

This laboratory is an introduction to the use of symbolic mathematics in MATLAB. It covers the creation of symbolic objects, the plotting of symbolic functions, the solution of algebraic and differential equations.

The steady state and transient response of First Order circuits is reviewed (see appendix), and the response of Second Order Circuits is introduced. A level of knowledge is assumed equivalent to ELEC1102 Foundations of Electronic Circuits or ELEC1103 Fundamentals of Electrical and Electronic Engineering.

References regarding circuit analysis are to "Introduction to Electric Circuits", 6th edition, Wiley, 2004, by Dorf and Svoboda (DS) and "Electrical Engineering – Principles and Applications", 4th edition, Pearson, 2008, by A R Hambley (ARH).

Symbolic Objects

The Symbolic Maths Toolbox in MATLAB uses a special data type called a **symbolic object** which can be used to represent **symbolic variables**, **numbers** and **expressions** which can be manipulated symbolically. The toolbox incorporates the kernel from the software package MAPLE.

Getting Help

Help on functions of the Symbolic Maths Toolbox can be obtained in the usual ways. Some functions, such as diff, however, exist in a numeric and symbolic form. The command

```
>> help diff
```

gives help on the numeric version. For the symbolic version, use the command

```
>> help sym/diff
```

There is also a command mhelp to get help about MAPLE functions.

Creating Symbolic Variables

Symbolic variables and expressions can be created using the sym function, for example:

```
>> x = sym('x'); a = sym('alpha')
```

creates two variables x and a which display as x and alpha respectively. If the variable name and the display string are the same, it is simpler to use syms. Thus

```
>> syms x y z
```

creates the three symbolic variables x, y and z.

Variables created in this way are complex. A symbolic variable can be specified to be real. To create real variables x and y:

```
>> x = sym('x', 'real'); y = sym('y', 'real');
```

or, simply

```
>>syms x y real
```

Symbolic Numbers

Numbers can also be represented symbolically using sym. An example of converting to rational number form (other forms are possible):

```
>> x = sym(0.1)  % sym(0.1, 'r') does the same thing
x =
1/10
>> y = sym(sqrt(5))
y =
sqrt(5)
```

By default, the numbers are represented in rational form if such a form exists. (√5 cannot be represented in this way). Calculations can be carried out, using such representations, with infinite precision.

Other symbolic representations of a number are possible. For example

```
>> y = sym(sqrt(5),'d')
y =
2.2360679774997898050514777423814
```

is the value of √5 to 32 decimal digits. The number of digits depends on the setting of digits (default 32). To set the value to 10 digits:

```
>> digits(10)
>> y = sym(sqrt(5),'d')
y =
2.236067977
```

The function double can be used to convert a symbolic number to its normal numeric value.

```
>> z = double(y)   % z is a double, not a sym object
z =
   2.2361
```

Symbolic Expressions and Functions

We can create symbolic expressions and functions of symbolic variables, for example

```
>> syms t a b
>> f = sin(a*t+b)
f =
sin(a*t+b)
```

and operations such as differentiation (see later) can be carried out on them. Note that f is automatically a symbolic object when it is defined in terms of symbolic variables.

For displaying symbolic expressions on the screen in a more readable form, the function pretty can be very useful. It displays expressions in a more readable form.

```
>> syms x
>> f = x^4+2*x^3-6*x^2+10
f =
x^4+2*x^3-6*x^2+10
>> pretty(f)
                    4     3     2
                   x  + 2 x  - 6 x  + 10
```

The Default Symbolic Variable

When an operation such as differentiation is performed on a function such as $\sin(at+b)$, it is often assumed that the differentiation should be carried out with respect to t. MATLAB does something similar. Thus, if the function `diff` is used to find the derivative of the function `f` above:

```
>> f = sin(a*t+b);
>> diff(f)
ans =
a*cos(b+a*t)
```

the answer is as one might expect. But suppose we want the derivative with respect to a.

```
>> diff(f, 'a')
ans =
t*cos(b+a*t)
```

In the first case of the derivative, MATLAB uses the following rule to decide on the **default symbolic variable**:

> The default symbolic variable in a symbolic expression is the letter closest to 'x' alphabetically. If two letters are equally close, the later one is chosen.

The `findsym` function uses this rule to find the default variable:

```
>> findsym(f, 1)
ans =
t
```

Plotting Functions

The `ezplot` function can be used to plot symbolic functions. For details, use `help sym/ezplot`. It will plot a function of one variable `f(x)` over the default range $= 2\pi < x < 2\pi$ with

```
>> ezplot(f)
```

The range can be specified if desired. It will also plot an implicitly defined function `f(x,y)` or a parametrically defined curve, i.e., `f(x,y)` with `x` and `y` functions of `t`, say.

The usual functions such as `axis`, `grid`, etc, can be used to affect the display.

Getting Solutions

Calculus

A number of functions are available for performing calculus: `diff` (differentiation), `int` (integration), `symsum` (summation) and `limit` (limits). See the documentation for details.

Simplification of Expressions

There are a number of functions which simplify expressions and carry out substitutions. They include `collect`, `expand`, `factor`, `simplify` and `simple`.

The `collect` function collects all coefficients of a polynomial with the same power:

```
>> f = 2*x-6+x*(x^2+2*x-7);
>> collect(f)
ans =
x^3+2*x^2-5*x-6
```

The `expand` function distributes products over sums and can do useful things to functions of sums:

```
>> f = a*(x+y);
>> expand(f)
ans =
a*x+a*y
>> expand(cos(x+y))
ans =
cos(x)*cos(y)-sin(x)*sin(y)
```

The `factor` function expresses a polynomial in terms of its factors (provided it can be done using rational numbers):

```
>> f = x^3+2*x^2-5*x-6;
>> factor(f)
ans =
(x+1)*(x-2)*(x+3)
```

The `simplify` function does general simplification including cancellation of common factors, application of trig identities, etc:

```
>> f = (x^3-3*x^2+3*x-1)/(x-1);
>> simplify(f)
ans =
x^2-2*x+1
>> simplify(sin(x)^2+cos(x)^2)
ans =
1
```

The `simple` function tries to produce the form of expression with the fewest number of characters (not always the most useful result).

In gaining familiarity with these functions, a willingness to experiment is most useful.

Substitutions

The function `subs` carries out symbolic substitution. For details, use `help sym/subs`.

```
>> f = (x+y)^2 + 3
f =
(x+y)^2+3
>> x = 5; y = z^3;
>> subs(f)
ans =
(5+z^3)^2+3
```

Alternatively,

```
>> subs(f, {x, y}, {sym('5'), z^3})
ans =
(5+z^3)^2+3
```

Other examples:

```
>> subs(f, 2)    % substitutes for default variable x
ans =
(2+y)^2+3
>> subs(f, y, (a+z))    % substitute for y
ans =
(x+a+z)^2+3
```

Solving Algebraic Equations

The solve function solves one or more equations for nominated variables. The simplest form simply accepts an expression as an argument, assumes the expression is equated to zero and solves for the default variable:

```
>> solve(a*x - b)
ans =
b/a
```

The command

```
>> solve('a*x = b');
```

performs exactly the same task. If you use the =, the quotes must be present! To solve for a instead of x:

```
>> solve(a*x-b, a)
ans =
b/x
```

We can solve simultaneous equations as follows:

```
>> S = solve('a + 2*b = 0', '3*a - b = 5')
S =
    a: [1x1 sym]
    b: [1x1 sym]
>> S.a
ans =
10/7
>> S.b
ans =
-5/7
```

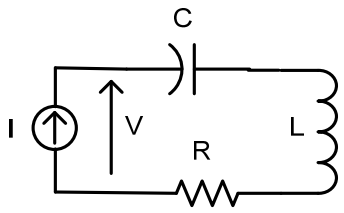
Here, the solution is returned in the structure `S` which has fields `a` and `b`. There are two equations and two variables in this example, so there is no ambiguity about which variables to solve for. In other situations, you may need to specify the variables:

```
>> S = solve(V - E1 + V/3/s + (V-E)*s/4, (E - V)*s/3 + E/4/s + E, V, E);
>> S.E
ans =
(48*E1*s^3)/(84*s^3+84*s+169*s^2+12)
```

Here, there are two equations in the four symbolic variables `V`, `E1`, `E` and `s`. A solution is carried out for `V` and `E` in terms of `E1` and `s`, returning the structure `S` containing `S.E` and `S.V`, and the result for `S.E` is displayed.

Steady State Sinusoidal Response of Second Order LCR Circuits

Please refer to the Appendix for a brief review of sinusoidal analysis with phasors. Text reference is DS Sec 13.5, ARH Sec 6.6.



A series LCR circuit is shown above.

Suppose the current source supplies a current $I = A\cos(\omega t)$ amps, then the phasor representation of V is

$$\mathbf{V} = \mathbf{IZ} = A \times \left(R + j\omega L - \frac{j}{\omega C} \right)$$

where the total complex impedance ***Z*** has been found by adding the impedances of each element.

The amplitude of ***V*** is

$$|\mathbf{V}| = A \times \left(R^2 + \left(\omega L - \frac{1}{\omega C} \right)^2 \right)^{1/2}$$

The reactance ωL of the inductor is positive and the reactance $-1/\omega C$ of the capacitor is negative, so there is a minus sign in the equation above. Consider the amplitude $|\boldsymbol{V}|$ as a function of ω. At low frequencies, $|\boldsymbol{V}|$ becomes very large because the reactance of the capacitor approaches infinity. At high frequencies, the same thing happens, because the reactance of the inductor becomes very large.

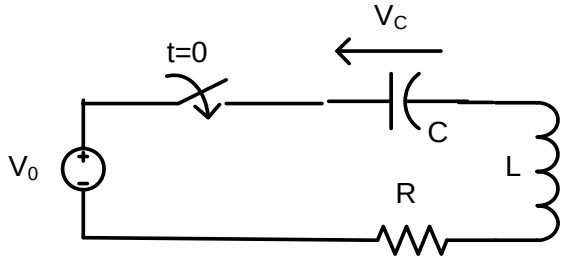
In between, there is a frequency where |**V**| has a minimum because the positive reactance of the inductor and the negative reactance of the capacitor cancel out. This is called the *resonant frequency*, or more precisely the *forced resonant frequency* since we are applying a sinusoidal forcing function $A\cos(\omega t)$ to the circuit. This frequency ω_0 is given by

$$\omega_0 L - \frac{1}{\omega_0 C} = 0 \quad \text{or} \quad \omega_0 = \frac{1}{\sqrt{LC}}$$

A similar resonance at the same frequency occurs for parallel LCR circuits, however, in that case, the impedance has a maximum value at the resonance rather than a minimum value as it does here.

Transient response of Second Order Circuits

The appendix revises First Order circuits. The text reference is DS Ch 9, ARH Sec 4.5. Just as circuits with a single irreducible energy storage element lead to a first order DE when calculating the transient response, circuits with two storage elements lead to a second order DE.



In the circuit above, a constant voltage V_0 is connected to a series LCR circuit at time $t = 0$. The resultant DE for $v_c(t)$ is (see Exercise 6):

$$V_0 = LC\frac{d^2 v_c}{dt^2} + RC\frac{dv_c}{dt} + v_c$$

Steady state response

The steady state (or forced response) is the value v_c will have after a long time. Since the forcing term is dc (the voltage V_0) the steady state value will also be dc, and the time dependent terms in the DE disappear, giving

$$v_c(\text{steady state}) = V_0$$

In mathematical terms, the steady state solution is a *particular* solution to the DE.

Transient response

For the transient or natural part of the solution, we set the constant forcing term to zero and try a solution of the form Ae^{st} by substituting in the DE:

$$LCs^2 + RCs + 1 = 0$$

This quadratic equation in *s* is called the *Characteristic Equation* of the DE. Notice that it is obtained by replacing *d/dt* in the DE with *s*, and d^2/dt^2 with s^2. Before solving, it is useful to introduce some general quantities which allow series and parallel LCR circuits to be analysed together:

	series circuit	Parallel circuit
damping factor α	$\frac{R}{2L}$	$\frac{1}{2RC}$
(Forced) resonant frequency ω_0	$\frac{1}{\sqrt{LC}}$	$\frac{1}{\sqrt{LC}}$

Then the solution to the Characteristic Equation for both series and parallel circuits becomes

$$s = -\alpha\left(1 \pm \sqrt{1 - \frac{\omega_0^2}{\alpha^2}}\right)$$

According as the discriminant is positive, negative or zero, there are three cases:

(a) When $\alpha > \omega_0$ the solution is two real negative values of s:

$$s_1 = -\alpha\left(1 - \sqrt{1 - \frac{\omega_0^2}{\alpha^2}}\right) \prec 0 \quad \text{and} \quad s_2 = -\alpha\left(1 + \sqrt{1 - \frac{\omega_0^2}{\alpha^2}}\right) \prec 0$$

The *transient* response becomes a sum of two decaying exponentials

$$v_c(\text{transient}) = k_1 e^{s_1 t} + k_2 e^{s_2 t}$$

And the *complete* response, with the *steady state* response added in, is

$$v_c(\text{complete}) = V_0 + k_1 e^{s_1 t} + k_2 e^{s_2 t}$$

The constants k_1 and k_2 are evaluated from the initial conditions at $t = 0$, using the values of the inductor current and the capacitor voltage. In the circuit here we will assume that the capacitor was initially discharged before $t = 0$. Also, since the switch is open before $t = 0$, the current through the inductor will be zero initially. This is a very common initial condition corresponding to no energy stored in the circuit at $t = 0$, either in the inductor or the capacitor. Since the current through the inductor is the same as the current through the capacitor:

$$i = C\frac{dv_c}{dt} = 0 \quad \text{at} \quad t = 0 \quad \text{, and} \quad v_c = 0 \quad \text{at} \quad t = 0$$

Substituting the expression for the complete response, we find

$$k_1 s_1 e^{s_1 t} + k_2 s_2 e^{s_2 t} = 0 \quad \text{at} \quad t = 0 \quad \text{, and} \quad V_0 + k_1 e^{s_1 t} + k_2 e^{s_2 t} = 0 \quad \text{at} \quad t = 0$$

So we have two linear equations for the constants k_1 and k_2 :

$$k_1 s_1 + k\ s_2 = 0 \quad \text{, and} \quad V_0 + k_1 + k_2 = 0$$

Solving,

$$k_1 = \frac{s_2 V_0}{s_1 - s_2} \quad \text{and} \quad k_2 = \frac{s_1 V_0}{s_2 - s_1}$$

This completes the solution. The above result should not be memorized. It is the method and form of the solution which are important. Notice that it is very similar to first order circuits, the only difference is that we have two decaying exponentials and two constants instead of one.

This type of solution is called ***overdamped***, because the large damping factor α dominates.

(b) When $\alpha<\omega_0$ the solution is two complex values of s, which are complex conjugates and have negative real parts:

$$s_1 = -\alpha + j\omega \quad \text{and} \quad s_2 = -\alpha - j\omega$$

where

$$\omega = \sqrt{\omega_0^2 - \alpha^2}$$

The final expression for v_c must be real, and using Euler's Relation, it can be shown:

$$v_c(\text{transient}) = Ae^{-\alpha t}\cos(\omega t + \phi)$$

and

$$v_c(\text{complete}) = V_0 + Ae^{-\alpha t}\cos(\omega t + \phi)$$

The constants now are A and ϕ, to be evaluated from the initial conditions at $t = 0$, again using the initial conditions. This type of solution is called ***underdamped***, because the damping factor α is small and the response oscillates with decreasing amplitude due the exponential (damping) term.

(c) When $\alpha=\omega_0$, the system is called **critically damped.** Special methods are required to treat this case since the Characteristic Equation has two equal roots because the discriminant is zero. We will not cover the solution here.

To describe the degree of damping, it is useful to introduce the *damping ratio* $\zeta = \alpha/\omega_0$, which is less than 1 for underdamped, 1 for critical damping, and greater than 1

for overdamped. For design purposes, the critically damped response can be useful, because in some senses it gives a response close to the voltage step applied by the voltage source and switch in the circuit (see Exercise 5).

Exercises

Exercise 1

Use help to view documentation on the functions diff, symsum, int and limit and then create the relevant symbolic variables and functions to determine the following:

$$\frac{d}{dt}\exp(-at)\sin(bt+c)$$

$$\lim_{x\to 2^-}\frac{5}{x-2}, \quad \lim_{x\to 0}\frac{5}{x-2}$$

$$\int_0^{\pi/2}\frac{1}{1+\tan x}dx$$

$$\sum_{k=1}^{20}\frac{1}{(-1)^{k-1}k^2} \text{ (to 4 decimal places)}$$

Exercise 2

Use solve to find the all the solutions for x and y to 4 decimal places if

$$x^2+10xy+4y^2=15$$
$$y=2x+1$$

Exercise 3

Forced response of LCR circuits

A parallel LCR circuit with inductance 20mH, capacitance 1μF and resistance 1kΩ is connected to a sinusoidal current source supplying 10A at frequency ω.

(a) Use complex admittances with symbolic maths to find an expression for the phasor voltage, and ezplot the magnitude of the voltage from ω=100 rad/s to 20000 rad/s. Observe the resonance and check its location. Explain the magnitude of the voltage at resonance. Decrease R and plot again. What has happened?

(b) Plot the phase angle of the voltage over the same range. Note: since angle does not accept symbols, you can generate an array of voltage values, take the angle, and use plot. Notice that there is a phase change of 180° through resonance. This feature arises because at low frequencies, the inductor dominates and the voltage leads the current by 90°, while at high frequencies the capacitor dominates and the voltage lags by 90°, passing through zero phase difference at resonance. This is a standard feature and occurs for series circuits also, with the direction of the phase change depending on the circuit details.

Exercise 4

Natural response of LCR circuits

We will investigate the series LCR circuit described in the notes. Suppose that L=1H, C=1F, V_0=1V, and the damping ratio ζ is unknown.

(a) Write script in an m-file that (i) calculates the value of R in terms of ζ, and (ii) sets up and solves the characteristic equation to find two values of s. Test the script for some ζ values by substituting in the final expression. (Hints: zeta is already defined in Matlab, so don't use this name; use digits and vpa for improved display.)

(b) Add script with symbolic maths to generate the complete solution for the capacitor voltage in terms of two unknown constants, k_1 and k_2 , and time t. Use symbolic maths to apply the initial conditions, solve for k_1 and k_2, and substitute back into the complete solution. Test the script for some ζ and t values by substituting in the final expression. Explain what happens when ζ = 1. (Hint: final results for capacitor voltage should have no imaginary part, but there may be a small round-off error. This can be removed with real().)

(c) Turn the script into a function taking two-dimensional arrays of t and ζ as input. Determine the dimensions of the t array, and assume the ζ array has the same dimensions. Used nested for loops to calculate an output array of capacitor voltage for each pair of t and ζ values with the same dimensions.

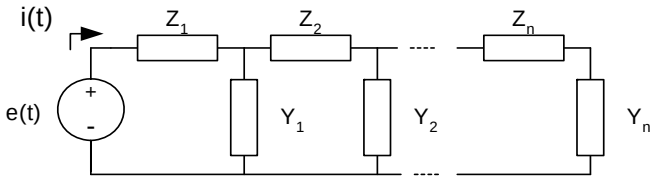
(d) Use meshgrid and mesh to plot the voltage over t=0s to 15s and ζ=0.01 to 1.5, making sure ζ=1 does not appear in the mesh. By rotating the mesh plot, investigate

(i) *overshoot* – by what % the signal passes its final value. Notice that overshoot appears when ζ>1. What is the overshoot as ζ approaches zero?

(ii) *risetime* – when there is no overshoot, how quickly the response rises from 10% to 90% of its final value. Notice that the critically damped solution has the fastest rise time without overshoot. In practice, a small amount of overshoot may be traded for a faster rise time.

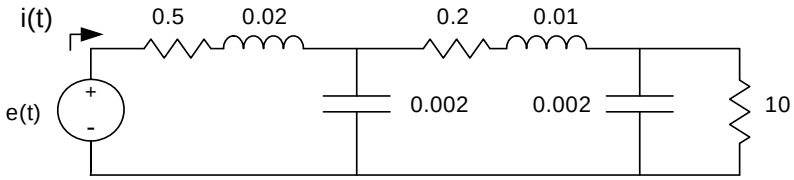
Exercise 5

This problem is concerned with steady state sinusoidal conditions in the ladder network shown.



Write a function zladder which accepts a vector of the values of the (phasor) impedances Z_1, Z_2, … in Ω and a vector of the values of the (phasor) admittances Y_1, Y_2, …in Ω^{-1} and returns the value of the total impedance in Ω connected across the voltage source. The impedances and admittances are permitted to have values of zero.

In the network below, values of resistances, inductances and capacitances are shown in Ω, F and H respectively. The voltage is $e(t) = 12\cos(100t + 30°)$ V. Using the function you have written, determine the total impedance and find $i(t)$. You may find it convenient to write a short script file that calls your function file.



Answers: Impedance = 1.2990 - 0.0947 Ω, $i(t) = 9.2137\cos(100t + 34.17°)$ A

Appendix - Review of Circuit Analysis

This section reviews some basic ideas of electric circuit analysis. Attention is confined to linear circuits containing independent voltage and current sources and ideal resistors, inductors and capacitors.

DC Analysis

A familiarity is assumed here with resistive circuits containing dc voltage and current sources. With such sources, inductors and capacitors are irrelevant to analysis, even if present: inductors appear as short-circuits and capacitors as open-circuits.

Two general methods for analyzing circuits are Node Voltage and Mesh Current analysis with basic steps as follows:

Mesh Current analysis

(a) Find the meshes in the circuit and assign a clockwise mesh current to each

(b) Write down Kirchhoff's Voltage Law (KVL) clockwise for each mesh with Ohm's Law for resistors. Going in the same direction as the current through a resistor makes a +ve contribution, going from – to + on a voltage source makes a –ve contribution.

(c) Solve for the mesh currents

Node Voltage Analysis

(a) Choose a reference node (usually at the bottom) and introduce a node voltage with respect to the reference for each of the remaining essential nodes.

(b) Write down Kirchhoff's Current Law (KCL) for each non-reference node, adding the currents flowing inward, and using Ohm's Law.

(c) Solve for the node voltages

Transient Analysis (First Order)

A first-order circuit only has one energy storage element (an inductor or capacitor) after simplification. If the sources in the circuit are dc, the solution for any circuit variable can always be written as a first order DE:

$$\frac{d}{dt}x(t)+\frac{x(t)}{\tau}=\frac{K}{\tau}$$

where τ is the time constant, either RC or L/R , and K is the *steady-state* or forced or final value of the variable $x(t)$, which depends on the sources in circuit.

The solution to the DE is

$$x(t)=K+[x(0)-K]\,e^{-t/\tau}$$

Where $x(0)$ is the initial value of the variable x. Behaviour is thus always a damped exponential approach from $x(0)$ to the final value K. The first (constant) term on the right is called the forced response since it arises from the sources in the circuit. The second (exponential decay) term is called the *transient* or *natural* response. The two terms added together are called the *complete* response of the circuit.

The solution to first order DE's is discussed from p484 in Palm (2005), where the second term is called the *free* response (as opposed to *forced*).

Steady State Sinusoidal Analysis

In an important class of problems, sources vary sinusoidally with time. We may, for example, have a voltage source

$$v(t) = V_m \cos(\omega t + \phi)$$

With such sources, inductors and capacitors cannot be ignored. The responses (voltages and currents) in the network will also be sinusoidal at the same frequency ω. Such problems are conveniently handled using **phasors** (DS 10.6, ARH 5.2). The phasor **V** representing $v(t)$ is essentially a complex number which contains information about the magnitude and phase of the voltage. Mathematically, we can write

$$v(t) = V_m \cos(\omega t + \phi) = \mathrm{Re}\left\{V_m e^{j(\omega t+\phi)}\right\} = \mathrm{Re}\left\{V_m e^{j\phi} e^{j\omega t}\right\} = \mathrm{Re}\left\{\mathbf{V} e^{j\omega t}\right\}$$

where

$$\mathbf{V} = V_m e^{j\phi} = V_m \angle \phi$$

is the phasor. If we generalise the idea of resistance to impedance and use phasors, many techniques for dc circuit analysis carry over to steady state sinusoidal analysis. The impedances of circuit elements can be summarised as follows:

a resistance of R ohms has an impedance of R ohms;

an inductance of L Henries has an impedance of $j\omega L$ ohms; and

a capacitance of C Farads has an impedance of $1/(j\omega C)$ ohms.

Rules for combining impedances in series, etc, are as for resistors in dc analysis.