Example Solution to a Second-Order Circuit Using the State-Space Method (employing \textit{matlab}'s \texttt{lsim} command)

Consider the circuit shown below (which is the same one given in the differential equation example):

Given the initial current through the inductor $i_1(t=0^-)$ is zero and the initial voltage across the capacitor is $v_c(t=0^-)$ is 10 V, find the current $i_2(t)$ for $t \geq 0$ using the state-space method. Additionally, to make things a bit more interesting, obtain the voltage across the 2 \Omega resistor which we will call $v_r(t)$. Let the forcing function be the ramp $v_g(t) = 3t$.

Let us define the two state variables as the inductor current, which corresponds to $i_1(t)$, and the capacitor voltage $v_c(t)$. Thus the state vector $x(t)$ is given by

$$
\begin{bmatrix}
i_1(t) \\
v_c(t)
\end{bmatrix}.
$$

After the switch has closed, the loop equation for the first loop is

$$
2i_1 + \frac{di_1}{dt} + v_c = v_g.
$$

Solving (2) for $\frac{di_1}{dt}$ yields

$$
\frac{di_1}{dt} = -2i_1 - v_c + v_g.
$$

This is one of the necessary state equations.
Since the voltage across the 5Ω resistor is the same as the voltage across the capacitor, the resistor current is given by

\[ i_2(t) = \frac{1}{5} v_c(t). \] (4)

Thus the desired output variable \( i_2 \) is related to one of the state variables in a rather trivial way. The other output variable is the voltage across the 2Ω resistor, \( v_r(t) \). Since the current through the inductor is the same as the current through this resistor, the voltage is given by

\[ v_r(t) = 2i_1(t) \] (5)

To obtain the second state equation, write the equation for the current through the capacitor:

\[ \frac{1}{3} \frac{dv_c}{dt} = i_1 - i_2 = i_1 - \frac{1}{5} v_c(t), \] (6)

which can be rewritten as

\[ \frac{dv_c}{dt} = 3i_1 - \frac{3}{5} v_c(t). \] (7)

Equations (3) and (7) allow us to write the complete state equation

\[
\begin{bmatrix}
\frac{di_1(t)}{dt} \\
\frac{dv_c(t)}{dt}
\end{bmatrix} =
\begin{bmatrix}
-2 & -1 \\
3 & -\frac{3}{5}
\end{bmatrix}
\begin{bmatrix}
i_1(t) \\
v_c(t)
\end{bmatrix} +
\begin{bmatrix}
1 \\
0
\end{bmatrix} v_g(t),
\]

(8)

\[ \frac{dx(t)}{dt} = A x + B v_g(t). \] (9)

Thus the matrices \( A \) and \( B \) are given by

\[ A = \begin{bmatrix} -2 & -1 \\ 3 & -\frac{3}{5} \end{bmatrix}, \quad \text{and} \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \] (10)

where, in this particular case, \( B \) is really a two-element column vector.

The initial-state vector, \( x_0 \) is given by

\[ x_0 = \begin{bmatrix} i_1(0) \\ v_c(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 10 \end{bmatrix}. \] (11)

The output equation is merely a combination of (4) and (5), neither of which depends on the input voltage \( v_g(t) \). Nevertheless, \( v_g(t) \) is included to conform to the standard form of the equation (but it is prevented from contributing anything by setting to zero the coefficients which multiply it)

\[
\begin{bmatrix}
i_2(t) \\
v_r(t)
\end{bmatrix} =
\begin{bmatrix}
0 & \frac{1}{5} \\
2 & 0
\end{bmatrix}
\begin{bmatrix}
i_1(t) \\
v_c(t)
\end{bmatrix} +
\begin{bmatrix}
0 \\
0
\end{bmatrix} v_g(t),
\]

(12)

\[ y = C x + D v_g(t). \] (13)
where matrices $C$ and $D$ are given by

$$
C = \begin{bmatrix}
0 & \frac{1}{2} \\
2 & 0
\end{bmatrix}, \quad \text{and} \quad D = \begin{bmatrix}
0 \\
0
\end{bmatrix}.
$$

(14)

As was the case with $B$ in the state equation, here $D$ is a two-element column vector (and happens to be zero).

**matlab Solution Using lsim**

The *matlab* command `lsim` can be used to obtain the solution to a problem which has been cast in state-space form. Until you have a good grasp of the `lsim` command, you may not find the description given by typing “`help lsim`” very helpful. For our purposes, `lsim` will typically be invoked as

```
lsim(a,b,c,d,u,t,x0)
```

or

```
[y,x] = lsim(a,b,c,d,u,t,x0)
```

In the first form, *matlab* will plot the state variables over the range of times specified by the vector $t$. The variables $a$, $b$, $c$, and $d$ contain the matrices with the corresponding names in the state and output equations. The variable $x0$ is a vector corresponding to the initial states. Finally, $u$ will have as many columns as there are inputs. Each row of $u$ corresponds to a new time point. The number of rows of $u$ must therefore correspond to the length of $t$. In the second form of the `lsim` command, where there are left-hand arguments present, $y$ will be set to the output response and $x$ will be set to the state response.

Let us plot the output over the first seven seconds. Thus $t$ will go from 0 to 7 and we will take increments of a tenth of second. We can define $t$ in *matlab* by

```
t = 0:.1:7
```

The final step before putting `lsim` to work is to specify the source term $u$. Given that we are told $v_g(t) = 3t$, $u$ is simply given by

```
u = 3*t
```

3
Here are the *matlab* commands that would give us the desired solution:

```matlab
>> a=[-2 -1; 3 -3/5]; b=[1; 0]; c=[0 1/5; 2 0]; d=[0; 0];
>> x0=[0; 10];
>> t = 0:.1:7; u=3*t;
>> lsim(a,b,c,d,u,t,x0)
```

Issuing these commands would produce the plots shown below:

The top plot shows the current $i_2(t)$ (since it was out first output variable, it is plotted first) and the bottom one is the voltage $v_r(t)$. Note the agreement with the solution we obtained using the classic differential equation approach. But, hopefully you recognize that state-space solution was much easier to obtain (especially when we let *matlab* do the work for us!).