Note 4b: State-Space Representation of Circuits

1 Introduction

In this note we introduce a simple technique that enables one to derive the state-space representation of any LCR circuit containing an dependent and independent sources. For simple circuits containing only a few elements, it's usually a matter of applying KCL/KVL and no formal procedure is needed. However, as the circuit becomes more complex, a systematic approach is needed.

While the reader can find an efficient approach to solving this problem in the classic literature, for example [1], the formulation requires knowledge of network theory beyond what we've covered so far. Here we present a simpler approach based on the simple repeated application of Thevenin and Norton's theorems.

The technique is best explained with an example circuit, shown in Figure 1, a two-section ladder LC low-pass filter.

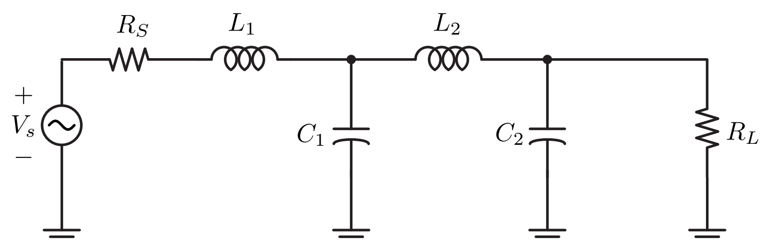


Figure 1: An example LCR circuit containing four state variables, the voltages of capacitors and currents of inductors.

2 The State of a Circuit

First we need to introduce the concept of state, more specifically the state of the circuit. The state of the circuit is the required knowledge to predict the future voltages/currents of a circuit. We known for a simple RC circuit, for example, knowledge of the capacitor voltage at a given time, say $t = t_0$, allows us to predict the voltages/currents in the RC circuit in the future, even if there are independent voltage / current sources. Elements such as resistors and dependent sources do not contribute to the state of the system because they have no memory and their $I - V$ relationships are instantaneous. On the other hand, capacitors, inductors, and other dynamic elements have memory and need to be accounted for if we are to find the future state of a circuit.

For this reason, we choose the state vector $\vec{x}$ to consists of the capacitor voltages and the inductor currents[1]:

$$\vec{x} = \begin{bmatrix} V_{C_1} & V_{C_2} & \cdots & V_{C_n} & I_{L_1} & I_{L_2} & \cdots & I_{L_m} \end{bmatrix}^\top \tag{1}$$

Which means that a circuit with n capacitors and m inductors has a total state vector of dimension $(n+m) \times 1$.

[1]Equivalently, we could also use the charge / flux of capacitors/inductors.

Now, given that the states of the circuit are known, simply take the original schematic (Figure 1), replace all inductors and currents with pseudo independent sources, and redraw the circuit, as shown in Figure 2. Note that we now can find all other voltages and currents in the circuit at any given time if we know the voltages/currents in the capacitors and inductors.

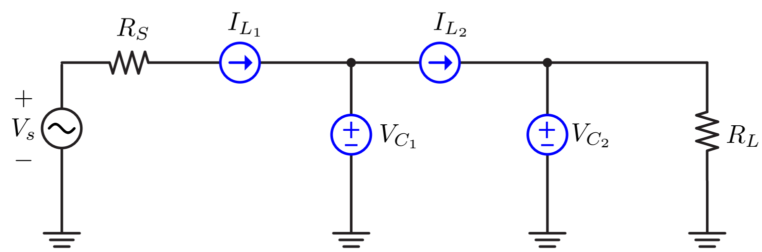


Figure 2: The equivalent circuit representation of the circuit presented in Figure 1. The state variables are represented with pseudo-sources (in blue).

The goal is to find a dynamic equation that describes the evolution of the state according to the following vector differential equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{x} = A\vec{x} + B\vec{b}_s \tag{2}$$

Where A is an $(n+m) \times (n+m)$ matrix and B is an $n+m \times k$ matrix, where k is the number of independent sources in the circuit. The source vector $\vec{b}_s$ is simply a list of all the independent sources in our original circuit.

Note that the state evolution only depends on the state variables $\vec{x}$ and the independent sources, and nothing else. The state vector for the example circuit is then

$$\vec{x} = \begin{bmatrix} V_{C_1} & V_{C_2} & I_{L_1} & I_{L_2} \end{bmatrix}^\top \tag{3}$$

The order of state variables doesn't matter, it's simply important to keep track of the variables. The vector $\vec{x}$ then represents voltages and currents in capacitors/inductors of the circuit at any given time allowing us to predict all other voltages/currents of the entire circuit.

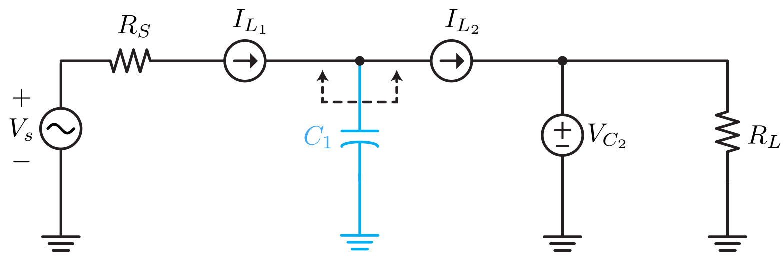
How can we systematically find the matrix A so that we can describe the evolution of the state?

3 Evolution of State Variables

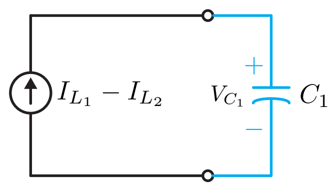
Let's start with the first state variable, which corresponds to the first row of the matrix A. The first row can be written as

$$\frac{\mathrm{d}V_{C_1}}{\mathrm{d}t} = A_{11}V_{C_1} + A_{12}V_{C_2} + A_{13}I_{L_1} + A_{14}I_{L_2} + B_{11}V_s \tag{4}$$

Our goal is to find the capacitor current ($C_1\frac{\mathrm{d}V_{C_1}}{\mathrm{d}t}$) as a linear combination of state variables plus any sources. To do this, let's replace all elements except the first by state variables, as shown in the Figure 3a. Note that we can use KCL and KVL to find the current into the capacitor but since we are after a systematic approach, we're going to represent everything "seen" by the capacitor by its Thevenin or Norton equivalent, as shown in Figure 3b.



(a) Calculation of the Norton equivalent seen by capacitor C_1.



(b) Equivalent circuit that determines the state dynamics.

In this particular example, it's clear that a Norton representation is an ideal current source since the current flowing into the capacitors is simply the difference between the currents of the two inductors:

$$I_N = I_{L_1} - I_{L_2} \tag{5}$$

Since current sources are driving C_1, there's no need to find R_N as it's infinity, which means the equivalent circuit is simply a current source driving C_1, or

$$C_1 \frac{\mathrm{d}V_{C_1}}{\mathrm{d}t} = I_N = I_{L_1} - I_{L_2} \tag{6}$$

Dividing through by C_1 we can see that the first row of our matrix is simply given by

$$A_{11} = A_{12} = B_{11} = 0 \tag{7}$$

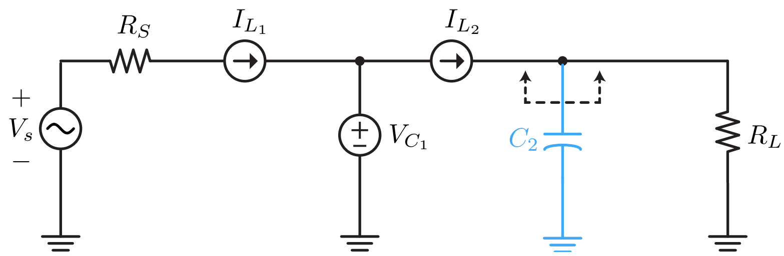
$$A_{13} = \frac{1}{C_1} \tag{8}$$

$$A_{14} = -\frac{1}{C_1} \tag{9}$$

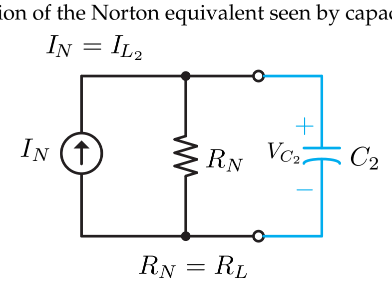
This process can be repeated for each state variable to find the complete matrix A. For example, to find the evolution of the second state V_{C_2}, we repeat the procedure and find the equivalent circuit shown in Figure 4a. The equivalent Norton circuit is used to deduce that

$$C_2 \frac{\mathrm{d}V_{C_2}}{\mathrm{d}t} = I_{L_2} - \frac{V_{C_2}}{R_L} \tag{10}$$

It's important to note that the right hand side only contains state variables and independent sources, which is true by construction in our approach.



(a) Calculation of the Norton equivalent seen by capacitor C_2.

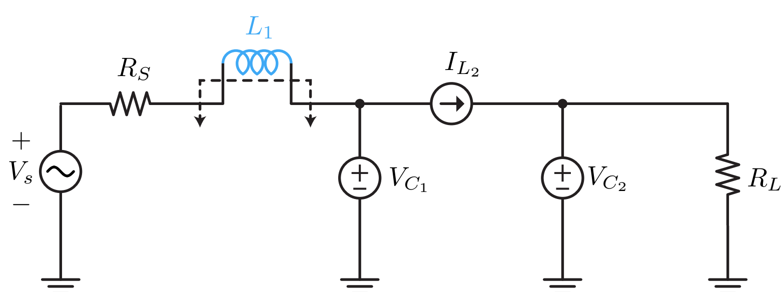


(b) Equivalent circuit that determines the state dynamics.

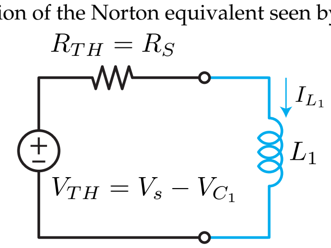
Let's now move to the first inductor. Again, drawing the equivalent circuit shown in Figure 5a, we use a Thevenin equivalent circuit to find

$$L_1 \frac{\mathrm{d}I_{L_1}}{\mathrm{d}t} = V_s - V_{C_1} - I_{L_1} R_S \tag{11}$$

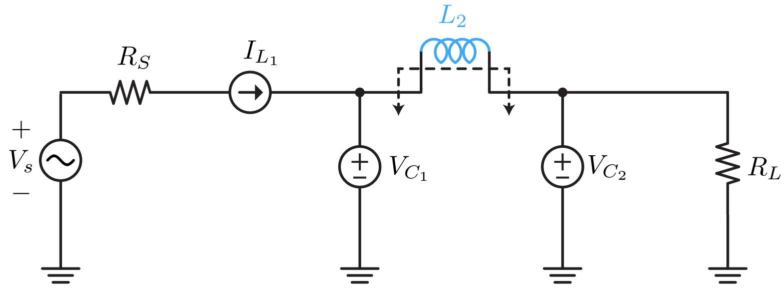
Here we see the independent source make an appearance, so $B_{31} = 1$. The other terms in the right hand side are state variables as desired.



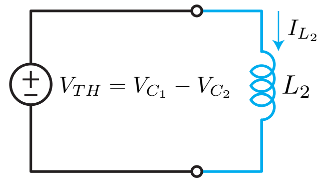
(a) Calculation of the Norton equivalent seen by inductor L_1.



(b) Equivalent circuit that determines the state dynamics.



(a) Calculation of the Norton equivalent seen by inductor L_2.



(b) Equivalent circuit that determines the state dynamics.

Finally, we come to the final state variable L_2, and follow the same procedure shown in Figure 6a and Figure 6b. We arrive at

$$L_2 \frac{\mathrm{d}I_{L_2}}{\mathrm{d}t} = V_{C_1} - V_{C_2} \tag{12}$$

This systematic procedure results in the following matrix A:

$$A = \begin{bmatrix} 0 & 0 & \frac{1}{C_1} & -\frac{1}{C_1} \\ 0 & -\frac{1}{R_L C_2} & 0 & \frac{1}{C_2} \\ -\frac{1}{L_1} & 0 & -\frac{R_S}{L_1} & 0 \\ \frac{1}{L_2} & -\frac{1}{L_2} & 0 & 0 \end{bmatrix} \tag{13}$$

and the matrix $B = \begin{bmatrix} 0 & 0 & \frac{1}{L_1} & 0 \end{bmatrix}^\top V_s$.

4 General Procedure

We can generalize from the previous example and summarize how one can approach an arbitrary circuit as follows. Suppose we are given a circuit consisting of linear elements, resistors, capacitors and inductors, and any number of sources or dependent sources. We can find the state representation as follows:

1. For each state in the circuit, i.e. inductors and capacitors, remove it from the circuit and replace all other inductors and capacitors using independent sources. Inductors become current sources and capacitors become voltage sources. Keep in mind that these represent the state variables, and not truly independent sources. We shall call them pseudo-independent sources to distinguish them from the truly independent sources.

2. Next find the Thevenin or Norton equivalent circuit seen by the state variable. In some cases you must choose Norton or Thevenin, while in most situation both will work. To find the Norton/Thevenin

circuit you should first find the open or short circuit voltage/current, and then the resistance seen looking into the circuit when the sources are turned off. Note that if dependent sources are present, you'll need to apply a test source.

3. Note the Thevenin/Norton equivalent circuit will consist of a linear combination of the other state variables (the pseudo-independent sources) and the independent sources in the circuit.

4. Now that you have the Thevenin/Norton equivalent, connect the state variable (inductor or capacitor) to the circuit and write the corresponding first-order differential equation that describes the voltage across the state inductor or the current through the state capacitor. This differential equation will be by design a linear combination of the other state variables and the sources. If R_T or R_N is finite, then the right hand side will also contain the state variable itself. The four cases are illustrated in Figure 7.

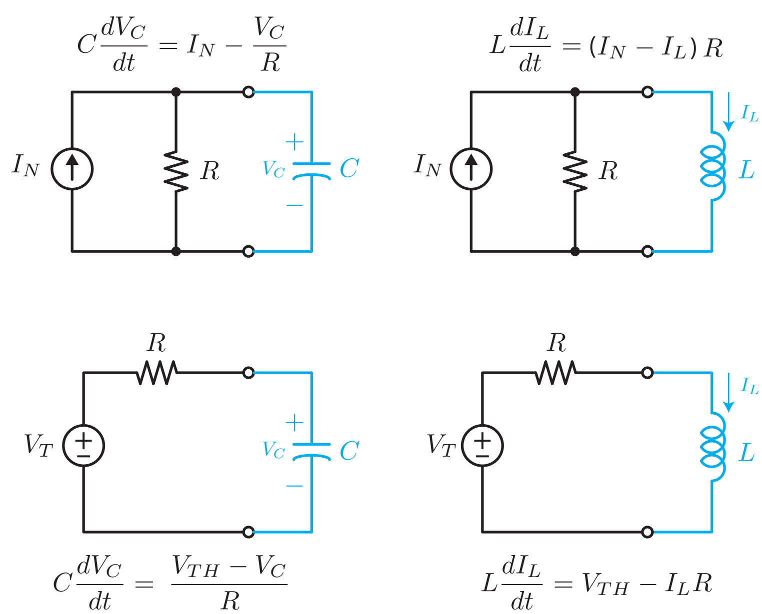


Figure 7: The four possible state variable dynamic equivalent circuit using application of Thevenin or Norton equivalent circuits.

This general procedure will result in the desired state vector differential equation. For each dynamic state variable, we will have an equation in the following form:

$$\frac{\mathrm{d}x_k}{\mathrm{d}t} = \sum_{i=1}^{n+m} A_{ki}x_i + \sum_{i=1}^{k} B_{ki}b_{s,i} \tag{14}$$

Grouping these state equations together results in the desired vector differential equation shown in eq. (2).

5 Numerical Example

We'll use Mathematica to do the calculations. Start by entering the matrix A we calculated above:

A = {{0,0,1/C1,−1/C1},{0,−1/(RLC2),0,1/C2},{−1/L1,0,−RS/L1,0},{1/L2,−1/L2,0,0}};

We verify the matrix we entered is correct:

A//MatrixForm

$$\begin{pmatrix} 0 & 0 & \frac{1}{\text{C1}} & -\frac{1}{\text{C1}} \\ 0 & -\frac{1}{\text{C2RL}} & 0 & \frac{1}{\text{C2}} \\ -\frac{1}{\text{L1}} & 0 & -\frac{\text{RS}}{\text{L1}} & 0 \\ \frac{1}{\text{L2}} & -\frac{1}{\text{L2}} & 0 & 0 \end{pmatrix}$$

Let's enter the numerical constants. Here we set the parameters to design a 100 MHz corner frequency low-pass filter matched to $50\,\Omega$:

L1 = 7410^ − 9; L2 = 114.210^ − 9; C1 = 45.610^ − 12; C2 = 29.5910^ − 12; RS = 50; RL = 50;

Let's see the eigenvalues for this matrix:

Eigenvalues[A]//MatrixForm

$$\begin{pmatrix} -1.9034 \times 10^8 + 7.2085 \times 10^8 i \\ -1.9034 \times 10^8 - 7.2085 \times 10^8 i \\ -4.8545 \times 10^8 + 2.82637 \times 10^8 i \\ -4.8545 \times 10^8 - 2.82637 \times 10^8 i \end{pmatrix}$$

As expected, all eigenvalues have a negative real part, since the circuit is stable. Also note that there are 2 real eigenvalues, which have a decaying response, and a pair of complex conjugate eigenvalues, which correspond to sinusoidal decay, something we saw in second-order *LCR* circuits.

Now form the eigenvector matrix. Note that Mathematica returns the eigenvectors as a row vectors, so we need to transpose the result:

Q = Transpose[Eigenvectors[A]];

The following matrix, *Qi*, is the transformation matrix that moves a state vector to the eigenspace basis:

Qi = Inverse[Q];

Let's do a sanity check and form a diagonal matrix using the known eigenvectors and then we subtract off the expected diagonal matrix. The result should be the zero matrix.

Abs[Qi.A.Q − DiagonalMatrix[Eigenvalues[A]]]//Total//Total

0.0000110833

We see the resulting matrix is indeed very small. From the diagonal eigenvalue matrix:

Dlam = DiagonalMatrix[Eigenvalues[A]];

We need to specify the initial conditions. The following state vector corresponds to zero currents/voltages except for the first capacitors, which is charged to 1 V:

xi = {{1}, {0}, {0}, {0}};

Now we can solve the system in the eigenspace domain. We'll use the Matrix Exponential function which does just the right thing for a diagonal matrix. It forms an exponential for the diagonal elements and leaves the rest of the matrix unchanged (zero). This Matrix Exponential is actually capable of doing much more, but for now let's just focus on what we've learned and use it as a shorthand notation.

qsolve = MatrixExp[Dlam*t*].Qi.xi;

qsolve//MatrixForm

$$\begin{pmatrix} (0.+0.i)+(0.501869-0.158766i)e^{(-1.9034\times10^{8}+7.2085\times10^{8}i)t} \\ (0.+0.i)+(0.501869+0.158766i)e^{(-1.9034\times10^{8}-7.2085\times10^{8}i)t} \\ (0.+0.i)+(0.346811-0.101888i)e^{(-4.8545\times10^{8}+2.82637\times10^{8}i)t} \\ (0.+0.i)+(0.346811+0.101888i)e^{(-4.8545\times10^{8}-2.82637\times10^{8}i)t} \end{pmatrix}$$

Note the solution is a vector with four rows, each row describing the evolution of each eigenstate or *mode*, with characteristic time constant decay factor determined by the real part of the eigenvalue and the oscillatory part determined by the imaginary part. Next we move the solution from the eigen-basis to the state vector basis using the inverse transformation matrix, which is just Q:

vsolve = *Q*.qsolve;

Take a look at the solution for the first state variable:

vsolve[[1,1]]//FullSimplify

$(0.103695+0.125241i)e^{(-4.8545\times10^{8}-2.82637\times10^{8}i)t}+(0.103695-0.125241i)e^{(-4.8545\times10^{8}+2.82637\times10^{8}i)t}+(0.396305+0.125371i)e^{(-1.9034\times10^{8}-7.2085\times10^{8}i)t}+(0.396305-0.125371i)e^{(-1.9034\times10^{8}+7.2085\times10^{8}i)t}$

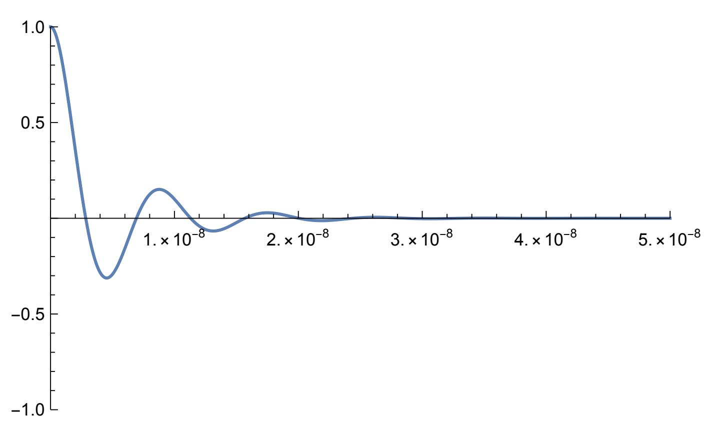
It's considerably more complicated than any of the modes (solution in the eigenspace) because it's a linear combination of the modes. Verify that all the state variables decay to zero:

Limit[vsolve, *t*->Infinity]//FullSimplify

$\{\{0.\},\{0.\},\{0.\},\{0.\}\}$

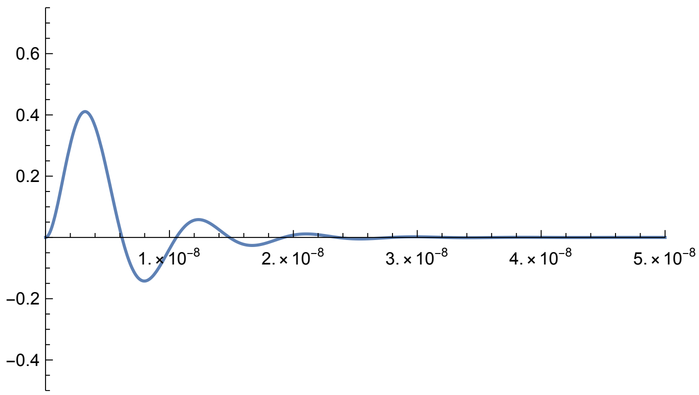
This must be true since we didn't excite the circuit with any sources (yet). Indeed, the state decays to zero, which is "obvious" if you replace all the inductors with short circuits and all the capacitors with open circuits. With $V_s = 0\,\text{V}$, the final output is zero. Finally plot the first state variable versus time, the voltages across the first capacitor:

Plot[vsolve[[1,1]], {*t*, 0, 5010^ − 9}, PlotRange->{{0, 5010^ − 9}, {−1, 1}}]



The plot shows that the voltage of the first capacitor decays to zero as it discharges and loses its charge to the load and source resistors. Due to the presence of the inductance, the capacitor actually swings back and forth, even charging negatively as it decays. Next, let's look at the second capacitor. Recall that the second capacitor was initially discharged.

Plot[vsolve[[2,1]], {*t*, 0, 5010^ − 9}, PlotRange->{{0, 5010^ − 9}, {−.5, .75}}]



We see the voltage rise from zero and hit a peak value of nearly 0.6 V before it too is discharged by the resistors. It's interesting to note that the first capacitor quickly charges up the second one on a faster time scale than the eventual decay and discharge.

Now let's add a voltage source driving the circuit to 1 V. We know that in steady state, the capacitors should charge to this value.

bs = {{0}, {0}, {1/L1}, {0}};

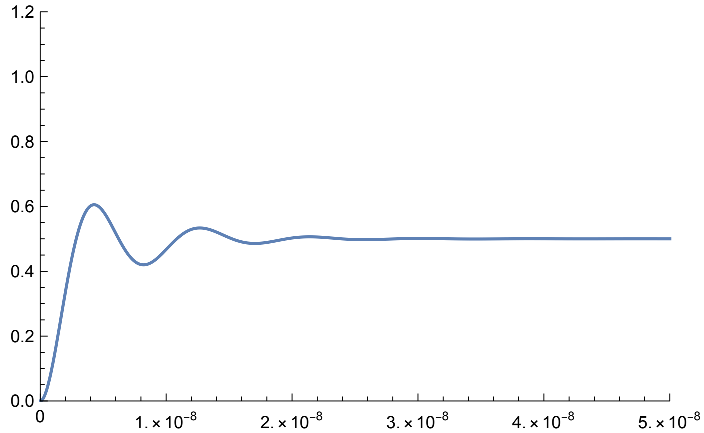
The complete solution consists of the homogeneous solution (just as before) plus the forced solution. Note that the forced solution is exactly the same as what we derived for the simple first-order differential equation using the integrating factor. The only difference is that the solution is now a vector equation. The operation Qi.bs first moves the source into the eigenspace basis, and each mode evolves according to the "weighted averaging" we saw earlier in first-order systems. Each state has a different weighing factor determined by the eigenvalue of the mode. Finally we move the solution from the eigenspace back to state space by multiplying by Q again. Make sure you understand all of these steps!

xforce[t_] = *Q*.MatrixExp[Dlam*t*].Qi.xi+

Q.MatrixExp[Dlam*t*].Qi.Integrate[*Q*.MatrixExp[−Dlam sigma].Qi.bs, {sigma, 0, *t*}];

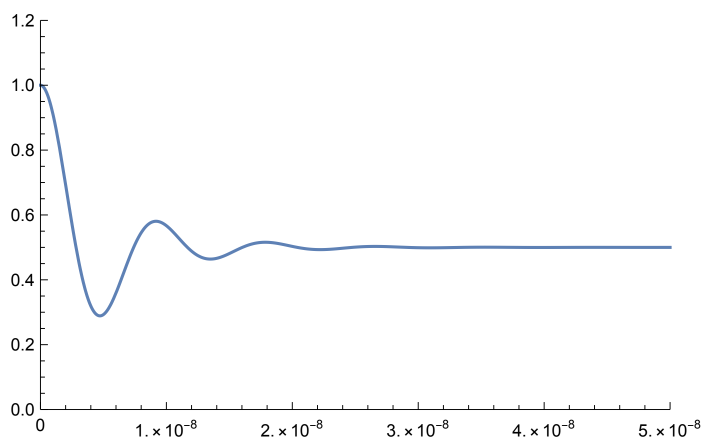
Plot the output voltage (second state variable) first to see the "step" response:

Plot[Re[xforce[*t*][[2]]], {*t*, 0, 5010^ − 9}, PlotRange->{{0, 5010^ − 9}, {0, 1.2}}]



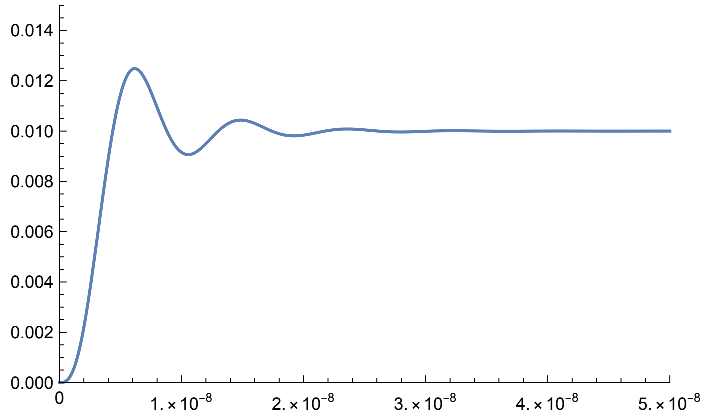
The capacitor is initially discharged and it charges by taking charge from both the first capacitor, just as before, but also now from the voltage source. This causes the voltage to rise quickly and stay high, unlike the previous case where it bounced down to a negative value. Note the steady state value is 0.5 V, as expected due to the voltage divider between R_s and R_L. Next let's examine the intermediate node voltage:

Plot[Re[xforce[t]][[1]]], {t, 0, 5010^ − 9}, PlotRange->{{0, 5010^ − 9}, {0, 1.2}}]

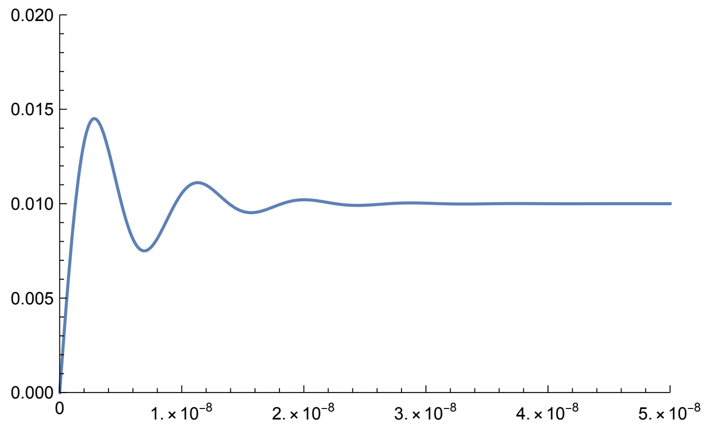


Recall that the capacitor is initially charged to 1V, so the value starts there. It initially dips as it loses charge to the first capacitor but then it recharges back to 0.5 V through the source. Finally let's see how the currents evolve. For the first inductor, we see a quick rise from 0 to the steady-state value:

Plot[Re[xforce[t]][[3]]], {t, 0, 5010^ − 9}, PlotRange->{{0, 5010^ − 9}, {0, 0.015}}]



Plot[Re[xforce[*t*][[4]]], {*t*, 0, 5010^ − 9}, PlotRange->{{0, 5010^ − 9}, {0, 0.02}}]



The second inductor also increases from zero and it overshoots the steady state value. This means that it's supplying more than the necessary current to drive the load, so the extra charge is going into the intermediate capacitor. Note that there's a delay in charging the first capacitor compared to the second.

Finally let's verify these voltages satisfy the steady-state conditions we expect:

Limit[*Q*.MatrixExp[Dlam*t*].Qi.Integrate[*Q*.MatrixExp[−Dlam sigma].Qi.bs, {sigma, 0, *t*}], *t*->Infinity]

{{0.5}, {0.5 + 0.*i*}, {0.01 + 0.*i*}, {0.01}}

The final values are also easily derived from the basic vector equation by setting all derivatives to zero:

−Inverse[*A*].bs

{{0.5}, {0.5}, {0.01}, {0.01}}

References

[1] Charles A Desoer and Ernest S Kuh, Basic Circuit Theory, McGraw-hill, 1969.

Contributors:

- Ali Niknejad.
- Anish Muthali.