Project 2: Transient Analysis and Simulation

The main objective of this project is to study how the transient response of a circuit can be simulated in Matlab and SPICE. As an illustration we will use the simple circuit in Fig. 5, in which $R_1 = R_2 = 2\,\Omega$, $R_3 = 1\,\Omega$, and $C = 0.5\,F$.

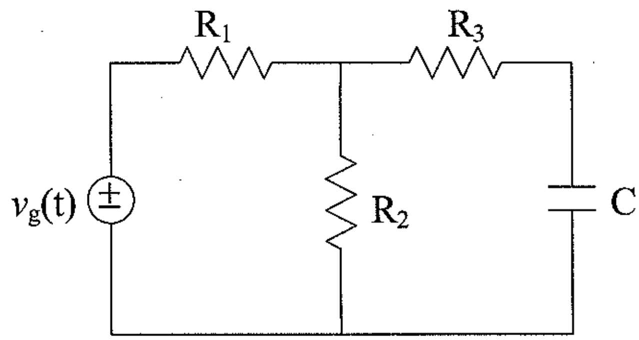

Fig. 5. A test circuit for transient analysis.

If we assume that $v_g(t)$ is a unit step function and that $v_C(0) = 0$, nodal analysis in the Laplace domain produces the following equations

$$
\begin{aligned}
&1)\ \ -I_{R1}(s) + I_{R2}(s) + I_{R3}(s) = 0 \\
&2)\ \ -I_{R3}(s) + I_C(s) = 0
\end{aligned}
\qquad
\begin{aligned}
I_{R1}(s) &= (V_g(s) - V_1(s))/R_1 \\
I_{R2}(s) &= V_1(s)/R_2 \\
I_{R3}(s) &= (V_1(s) - V_2(s))/R_3 \\
I_C(s) &= sCV_2(s)
\end{aligned}
\tag{23}
$$

After substituting the element values, (23) can be rewritten in matrix form as

$$
\begin{bmatrix} 2 & -1 \\ -1 & (1+s/2) \end{bmatrix} \begin{bmatrix} V_1(s) \\ V_2(s) \end{bmatrix} = \begin{bmatrix} 1/2s \\ 0 \end{bmatrix} \tag{24}
$$

Since this is a 2×2 matrix, we can invert it analytically, obtaining $V_1(s)$ and $V_2(s)$ as

$$
V_1(s) = \frac{1+s/2}{2s(s+1)} = \frac{1/2}{s} - \frac{1/4}{s+1} \tag{25}
$$

and

$$
V_2(s) = \frac{1}{2s(s+1)} = \frac{1/2}{s} - \frac{1/2}{s+1} \tag{26}
$$

The explicit expressions for $v_1(t)$ and $v_2(t)$ are now easily found to be

$$
v_1(t) = \frac{1}{2} - \frac{1}{4}e^{-t} \tag{27}
$$

and

$$
v_2(t) = \frac{1}{2} - \frac{1}{2}e^{-t} \tag{28}
$$

respectively.

It is important to recognize that this approach is not practical for large circuits. One of the main reasons for this is that some of the elements in the matrix are not numbers, but rather functions of s. This type of "symbolic" inversion is virtually impossible to perform for large matrices. In such cases the only alternative is *numerical simulation.*

As a preparatory step for such a simulation, it is necessary to describe the circuit as a combination of differential and algebraic equations (so-called DAEs). In our case, this implies rewriting the nodal equations as

$$\begin{array}{llll} & & & i_{R1}(t) = (v_1(t) - v_2(t))/R_1 \\ 1) \ i_g(t) + i_{R1}(t) = 0 & \Rightarrow & v_1(t) = v_g(t) & i_{R2}(t) = v_2(t)/R_2 \\ 2) \ -i_{R1}(t) + i_{R2}(t) + i_{R3}(t) = 0 & & & i_{R3}(t) = (v_2(t) - v_3(t))/R_3 \\ 3) \ -i_{R3}(t) + i_C(t) = 0 & & & i_C(t) = C\dot{v}_3(t) \\ & & & i_g(t) = ? \end{array} \tag{29}$$

These equations can be expressed in matrix form in the following way

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -C \end{bmatrix} \begin{bmatrix} \dot{v}_1(t) \\ \dot{v}_2(t) \\ \dot{v}_3(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1/R_1 & (1/R_1 + 1/R_2 + 1/R_3) & -1/R_3 \\ 0 & -1/R_3 & 1/R_3 \end{bmatrix} \begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \end{bmatrix} - \begin{bmatrix} v_g(t) \\ 0 \\ 0 \end{bmatrix} \tag{30}$$

Note that the first two equations in (30) are purely algebraic (there are no derivatives in them), while the last one is differential.

In order to solve system (30), we first need to define matrix

$$M = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -C \end{bmatrix} \tag{31}$$

and then enter the following three lines in Matlab:

$M = [0\ \ 0\ \ 0;\ 0\ \ 0\ \ 0;\ 0\ \ 0\ \ -0.5]$;

options=odeset('mass', M);

$[t, x]$=ode23t(@transient2, [0 5], x_0, options);

The function transient2.m is provided in Appendix 3, and the corresponding plots of $v_2(t)$ and $v_3(t)$ are shown in Fig. 6 (these are voltages accross R_2 and C, respectively).

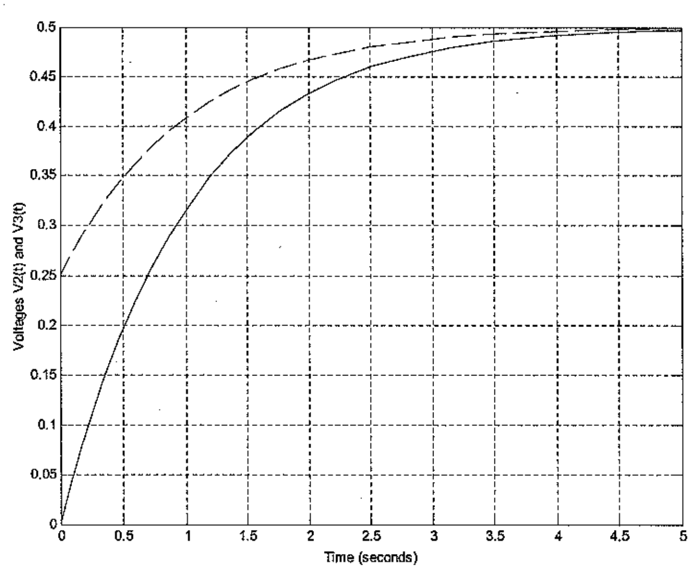

Fig. 6. The transient responses for the circuit in Fig. 5.

It is easily verified that these plots match the analytic expressions obtained in (27) and (28). We should also point out that SPICE uses a very similar approach in executing transient analysis.

NOTE: Since the step function is *discontinuous*, it cannot be handled numerically, and we must include a finite (although very small) rise time. Keep in mind also that the PULSE function in SPICE is *inherently periodic*. If you want to simulate a unit step, you must set the period to be *larger* than the overall simulation time.

Appendix 3

```
function F=transient2(t,x)
% This function provides the right-hand side of the
% differential equation for the Matlab solver.

F=[0;0;0];
% Vector F is initialized.

R1=2; R2=2; R3=1;
% These are the resistor values for our circuit. If you want
% to perform a simulation with different R1, R2 and R3, all you
% need to do is change this line.

if (t>=0)&(t<=1e-6)
   h=1e6*t;
else
   h=1;
end
% h(t) represents an approximation of the step function,
% with a rise time of 1 microsecond.

F(1)=x(1)-h;
F(2)=-(1/R1)*x(1)+(1/R1+1/R2+1/R3)*x(2)-(1/R3)*x(3);
F(3)=-(1/R3)*x(2)+(1/R3)*x(3);
% This is the right-hand side of the DAE. It is written in terms of R1, R2
% and R3, so that you don't need to rewrite the code every time your
% element values change.
```