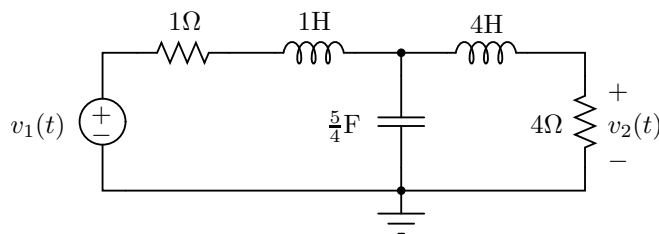
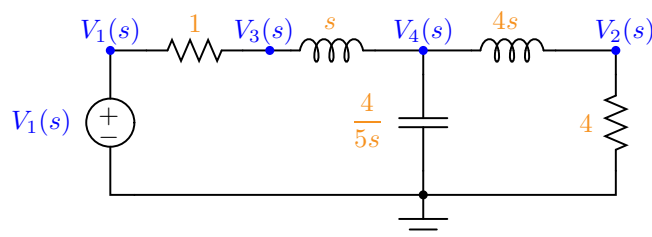


Circuit Analysis Using MATLAB

Consider the passive filter shown below. We will use this circuit to illustrate various MATLAB functions which are useful in the time- and frequency-domain analysis.

**Deriving the transfer function $H(s) = V_2(s)/V_1(s)$**

Let us first derive the transfer function $H(s) = V_2(s)/V_1(s)$. One way to accomplish this is to perform nodal analysis on a frequency-domain version of the circuit. If we redraw the schematic, in terms of impedances and s -domain variables, we get the following circuit.



It should be noted that there are only three unknown node voltages in the circuit, namely $V_2(s)$, $V_3(s)$ and $V_4(s)$ since $V_1(s)$ is assumed to be a known (source) voltage. We should be able to write three equations, in terms of the three unknown node voltages and solve for each of the variables in terms of the known voltage $V_1(s)$. Let us first write the nodal equations for the circuit using KCL at each node (sum of the currents leaving node equals zero). We will not write the “(s)” after the voltages to simplify the notation.

$$\begin{aligned} \text{KCL at node 2} &\Rightarrow \frac{V_2}{4} + \frac{V_2 - V_4}{4s} = 0 \\ \text{KCL at node 3} &\Rightarrow \frac{V_3 - V_1}{1} + \frac{V_3 - V_4}{s} = 0 \\ \text{KCL at node 4} &\Rightarrow \frac{V_4 - V_3}{s} + \frac{V_4 - V_2}{4s} + \frac{5s}{4}V_4 = 0 \end{aligned}$$

Now we are in a position to solve the equations and by now you should know at least a few different ways to do this. In this document, I will illustrate how to do this with MATLAB's Symbolic Toolbox. Note that the lines that begin with >> are user entries while % are comments. We have also used the semicolon ; at the end of most user entries (except the last one) to suppress the MATLAB output from appearing.

```
% Solve for a transfer function based on Nodal Analysis
% Declare symbolic variables
>> syms s V1 V2 V3 V4
% Nodal analysis equations for nodes 2, 3 and 4
>> eqn2=V2/4 + (V2-V4)/(4*s);
>> eqn3=(V3-V1)/1 +(V3-V4)/s;
>> eqn4=(V4-V3)/s+(V4-V2)/(4*s)+5*s*V4/4;
% Solve for V2, V3 and V4 in terms of V1
>> [V2,V3,V4]=solve(eqn2,eqn3,eqn4,V2,V3,V4)
V2 = (4*V1)/(5*s^3 + 10*s^2 + 10*s + 5)
V3 = (5*V1*s^3 + 5*V1*s^2 + 5*V1*s + 4*V1)/(5*s^3 + 10*s^2 + 10*s + 5)
V4 = (4*V1)/(5*s^2 + 5*s + 5)
```

Even though MATLAB provides expressions for all of the three unknowns, we are only interested in $V_2(s)$ as a function of $V_1(s)$. We can get this by looking at the first of three expressions that were produced by MATLAB. The transfer function $H(s)$ is given by

$$H(s) = \frac{V_2(s)}{V_1(s)} = \frac{4}{5s^3 + 10s^2 + 10s + 5}$$

By examining this transfer function, we note that the dc gain $H(j0) = 4/5$ while the high frequency gain $H(j\infty) \rightarrow 0$. Both of these observations make sense when compared to the structure of the circuit. It is clear that this circuit acts as a lowpass filter.

Finding the poles and zeros of $H(s)$

We find the poles and zeros of $H(s)$ by finding the roots of the denominator and numerator polynomials, respectively. Since the numerator is not a function of s there are no finite zeros of the transfer function; however, since the denominator is third order, there will be three zeros at infinity

```
% Numerator polynomial, N(s)
>> num=[4];
% Denominator polynomial, D(s)
>> den=[5 10 10 5];
% Find the zeros of H(s) by N(s)=0
>> roots(num)
ans =
    Empty matrix: 0-by-1
% Find the poles of H(s) by D(s)=0
>> roots(den)
ans =
    -1.0000
   -0.5000 + 0.8660i
   -0.5000 - 0.8660i
```

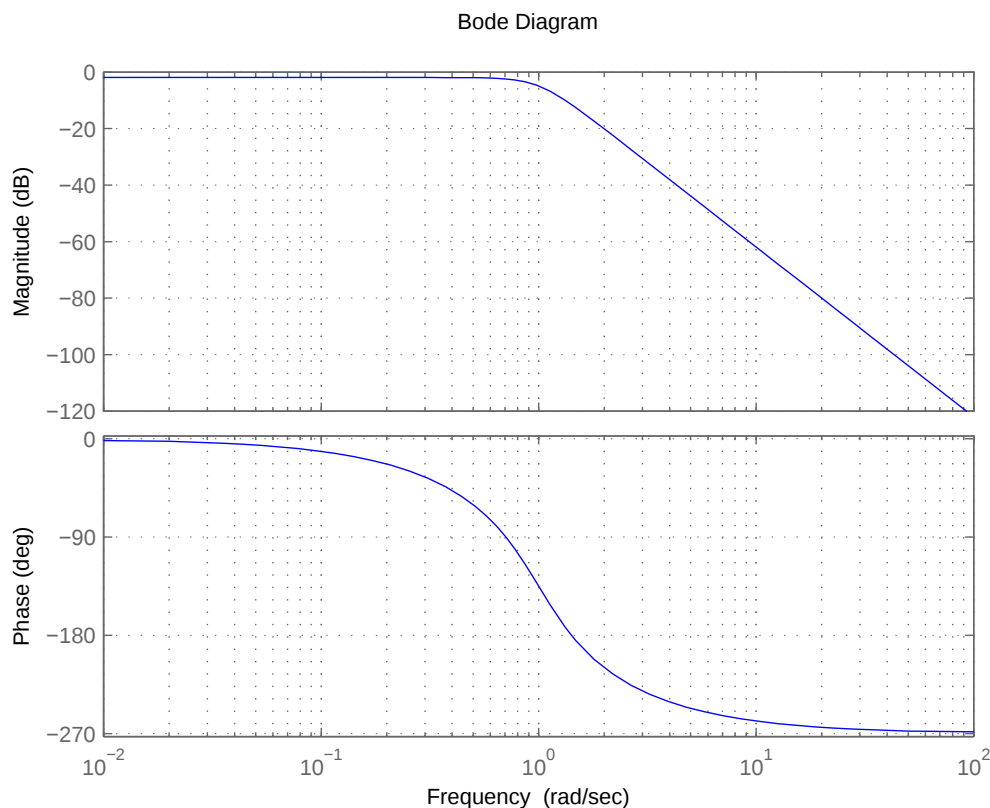
MATLAB output shows that there are no finite zeros; however, there are three poles: -1 and $-0.5 \pm j0.866$.

Finding the frequency response of $H(s)$

Determination of the frequency response is a very simple procedure in MATLAB and can be done by using the `bode` command. The first (and the easiest way) is to use the `bode` command without specifying a frequency range and let MATLAB figure out the best range for us. It should be noted that the `bode` command will simultaneously produce two plots in the same window: the magnitude in dB ($20 \log |H(j\omega)|$) and the phase ($\angle H(j\omega)$) both plotted versus frequency in rad/s (log-axis).

```
% Numerator polynomial, N(s)
>> num=[4];
% Denominator polynomial, D(s)
>> den=[5 10 10 5];
% Generate the Bode plot with grid
>> bode(num,den),grid
```

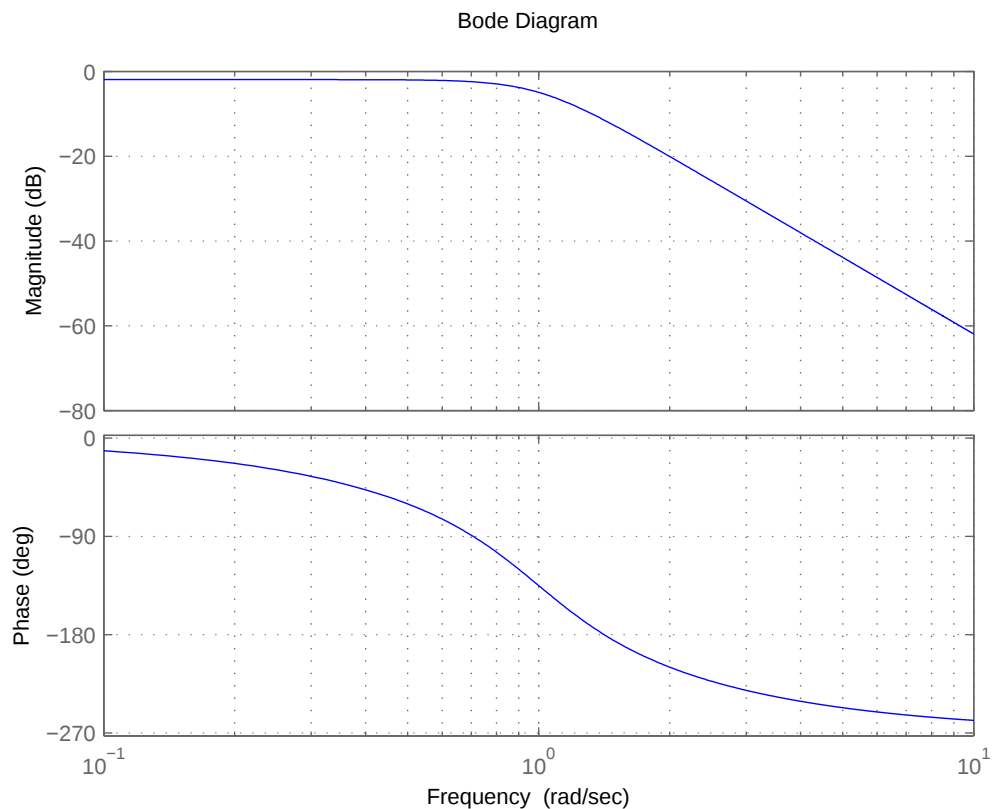
It can be seen from the figure that MATLAB has chosen the frequency range of 0.01 rad/s to 100 rad/s. If we want to change the frequency range or if we want to plot the linear magnitude $|H(j\omega)|$, we have to alter the way in which the `bode` command is issued.



Let us say that we would like to produce the frequency response over a narrower frequency range, for example from 0.1 rad/s to 10 rad/s. Here's how we can implement that change by using a slightly different version of the `bode` statement.

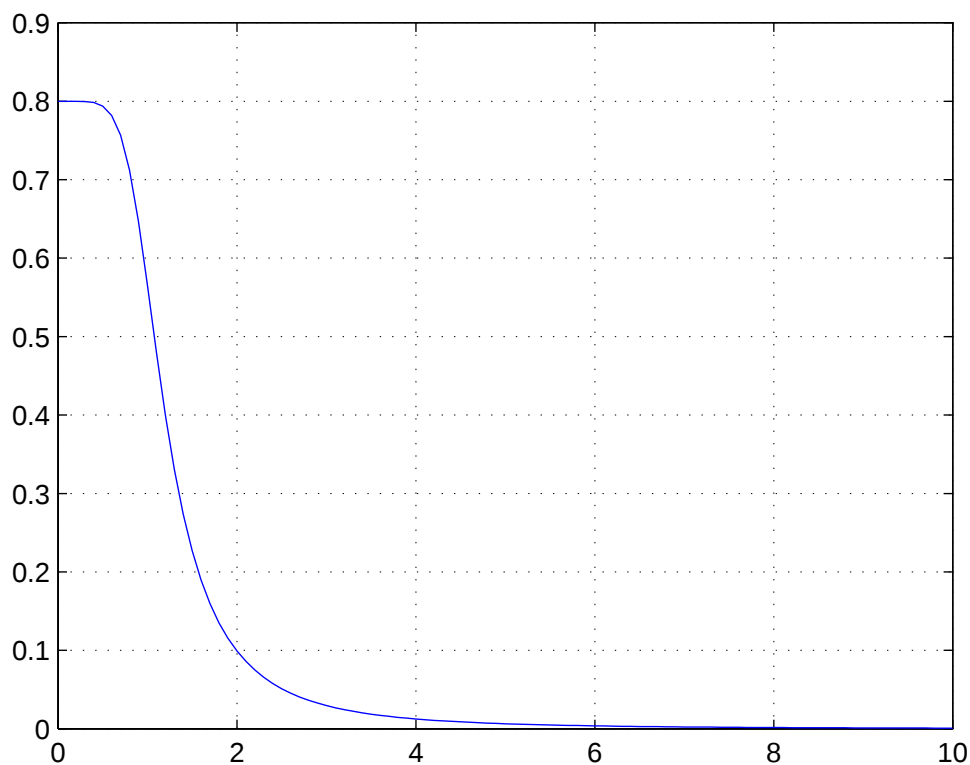
```
% Numerator polynomial, N(s)
>> num=[4];
% Denominator polynomial, D(s)
>> den=[5 10 10 5];
% Define frequency range from 0.1 rad/s to 10 rad/s in steps of 0.01 rad/s
>> w=0.1:0.01:10;
% Generate the Bode plot with grid
>> bode(num,den,w),grid
```

It can be seen from the figure that the frequency response now is shown in the frequency range from 0.1 rad/s to 10 rad/s.

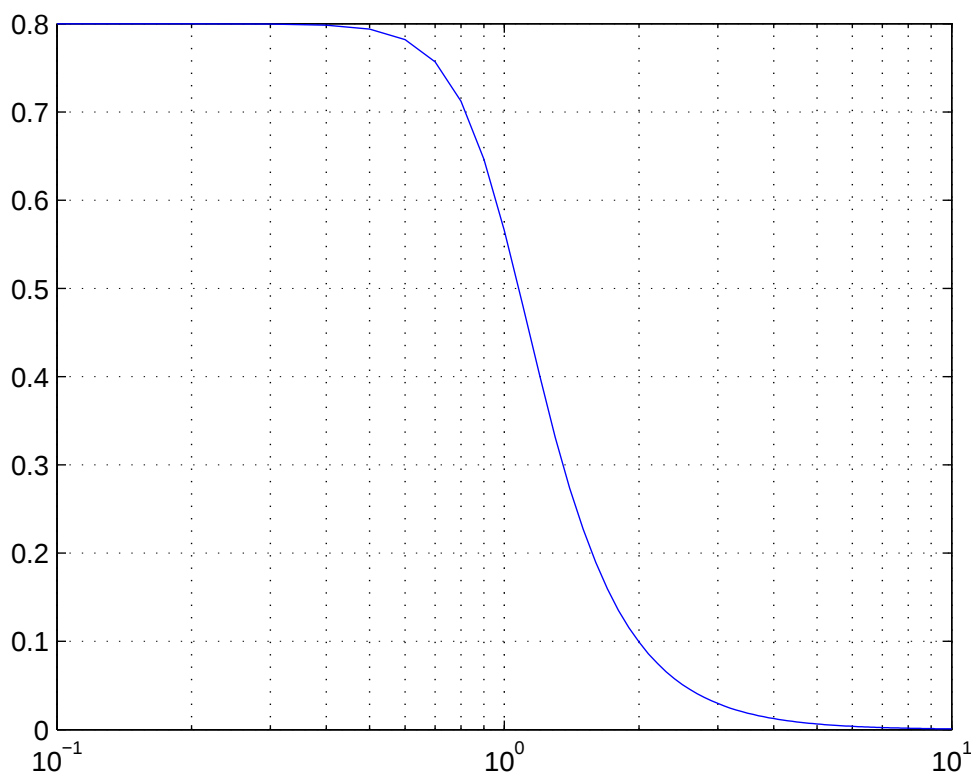


We can also use the `bode` command to plot the frequency response in a linear scale if we wish. For example, let us plot the frequency response of $H(s)$ from 0 to 10 rad/s.

```
% Numerator polynomial, N(s)
>> num=[4];
% Denominator polynomial, D(s)
>> den=[5 10 10 5];
% Define frequency range from 0 rad/s to 10 rad/s in steps of 0.1 rad/s
>> w=0:0.1:10;
% Generate magnitude and phase response without plotting and store
% them in variables called 'mag' and 'phase'
>> [mag,phase]=bode(num,den,w);
% Plot the linear magnitude response versus frequency (also linear)
>> plot(w,mag),grid
```



As can be seen from the figure, both the horizontal and the vertical axis are linear. If we want to have the horizontal axis be plotted in log mode instead we can simply use the command `semilogx` and re-plot the magnitude response with the command `semilogx(w,mag),grid` instead of the normal `plot(w,mag)` command. This will produce the plot shown next.



Note that if we issue the command `semilogx(w,20*log10(mag)),grid`, we will basically be plotting the magnitude response in dB – very similar to what we already saw when we used the `bode` command.

Finding the impulse and step responses of $H(s)$

The impulse and the step response are time-domain measures that are often used to characterize a system. When a system is described by its transfer function $H(s) = V_2(s)/V_1(s)$, the meaning of the impulse and step responses are very straightforward. First of all, it should be clear that, in the time-domain $v_1(t)$ is the excitation (input) and $v_2(t)$ is the response (output). Impulse response refers to $v_2(t)$ when $v_1(t) = \delta(t)$ (unit impulse) and the step response refers to $v_2(t)$ when $v_1(t) = u(t)$ (unit step). Recall that the unit step and the unit response are related by $\delta(t) = du(t)/dt$. And since the circuit is a linear, time-invariant, causal system, the responses are related in the same manner – the impulse response is the time derivative of the step response. Here's how we can quickly obtain the unit impulse and step responses for our system.

```
% Numerator polynomial, N(s)
>> num=[4];
% Denominator polynomial, D(s)
>> den=[5 10 10 5];
% Plot the impulse response with grid in window called Figure 1
>> figure(1); impulse(num,den),grid
% Plot the step response with grid in window called Figure 2
>> figure(2); step(num,den), grid
```

