Circuits II
EE221
Unit 4
Instructor: Kevin D. Donohue

Transfer Function, Complex Frequency, Poles and Zeros, and Bode Plots, Resonant Circuits
Node Voltage Example

- Perform phasor analysis to determine \( v_o(t) \). Since the source frequency is not specified, express impedances as \( j\omega = s \).
  - Derive phasor of \( v_o(t) \) in terms of a product between a rational polynomial in \( s \) and the phasor of the input. Then substitute \( \omega = 2 \) and solve for \( v_o(t) \).

\[
\begin{align*}
\hat{V}_o &= \hat{V}_s \frac{s}{s^2 + 3s + 3} = |\hat{V}_s| \frac{\sqrt{\omega^2}}{\sqrt{(3 - \omega^2)^2 + 9\omega^2}} \left( \angle(\hat{V}_s) + 90^\circ - \tan^{-1}\left( \frac{3\omega}{3 - \omega^2} \right) \right) \\
\end{align*}
\]

- Show \( v_o(t) = 0.33\cos(2t - 9.46^\circ) \) V, when \( \omega = 2 \)
Transfer Functions

- From the last example let:

\[ \hat{H}(s) = \frac{s}{s^2 + 3s + 3} \]

- Note that for \( s = j\omega \), the above function represents the phasor ratio of output to input for any \( \omega \):

\[ \frac{\hat{V}_o(j\omega)}{\hat{V}_s(j\omega)} = \hat{H}(j\omega) = \frac{\sqrt{\omega^2}}{\sqrt{(3-\omega^2)^2 + 9\omega^2}} \angle \left( 90^\circ - \tan^{-1}\left( \frac{3\omega}{3-\omega^2} \right) \right) \]

- Therefore, for any input magnitude, phase, and frequency, \( \omega_i \), the output can be determined by multiplying the input magnitude by \( |\hat{H}(j\omega_i)| \) and adding \( \angle \hat{H}(j\omega_i) \) to the input phase. In this sense, \( \hat{H}(s) \) describes how to transfer the input value to the output value.
Transfer Functions

- **Definition:** A transfer function (TF) is a complex-valued function of frequency associated with an input-output system, such that for a sinusoidal input, the TF evaluated at the input frequency is a complex number whose magnitude indicates the scaling between the input and output magnitudes, and phase indicates the phase shifting between the input and output phases.

- The transfer function indicates this relationship for any input frequency.

- To find a transfer function, convert to impedance circuit with impedance values expressed as functions of $j\omega = s$. Then solve for the ratio of the phasor output divided by the phasor input.
Transfer Function Example

Determine the transfer function for the circuit below, where the input is \( v_i(t) \) and the output is \( i_o(t) \).

Show transfer function is given by:

\[
\hat{H}(s) = \frac{\hat{I}_o}{\hat{V}_i} = \frac{1}{CR_1R_2} \left( \frac{R_1 + R_2}{s + \frac{1}{CR_1R_2}} \right)
\]
Transfer Function Example

Determine the transfer function for the circuit below, where the input is \( v_i(t) \) and the output is \( v_o(t) \).

Show transfer function is given by:

\[
\hat{H}(s) = \frac{\hat{V}_o}{\hat{V}_i} = \frac{1 + \frac{R_f}{R_i}}{s + \frac{1}{R_h C}}
\]
Evaluating/Plotting TFs

The following Matlab commands can be used to plot the results of the last TF. Let $R_f = 50 \Omega$, $R_i = 10k \Omega$, $R_h=1.59k \Omega$, $C = 1\mu F$

```matlab
Rf = 50e3; Ri = 10e3; Rh=1.59e3; C = 1e-6;
f = [0:500];  % generate x-axis points
s = j*2*pi*f;
h = (1+Rf/Ri)*s ./ (s + (1/(Rh*C)))); % Evaluate TF at every point
figure(1) % Plot Magnitude
plot(f,abs(h))
xlabel('Hertz')
ylabel('Magnitude')
figure(2) % Plot Phase in Degrees
plot(f,phase(h)*180/pi)
xlabel('Hertz')
ylabel('Degrees')
```
Resulting Plots

Magnitude

Hertz

Degrees

Hertz
TF Units

There are 4 possible ratios of voltages and currents that can be used for a TF. List all possibilities and indicate the units for each one.

\[
\hat{H}(j\omega) = \frac{\hat{V}_{out}(j\omega)}{\hat{V}_{in}(j\omega)}
\]

\[
\hat{H}(j\omega) = \frac{\hat{V}_{out}(j\omega)}{\hat{I}_{in}(j\omega)}
\]

\[
\hat{H}(j\omega) = \frac{\hat{I}_{out}(j\omega)}{\hat{V}_{in}(j\omega)}
\]

\[
\hat{H}(j\omega) = \frac{\hat{I}_{out}(j\omega)}{\hat{I}_{in}(j\omega)}
\]
Decibel Scale

A decibel is a logarithmic measure of gain (or attenuation). A power gain between the designated system input and output is denoted in Decibels (or dB) as:

\[ G_{dB} = 10 \log_{10} \left( \frac{P_{out}}{P_{in}} \right) \]

Express power in terms of voltage over a load to get another version of the dB formula:

\[ G_{dB} = 10 \log_{10} \left( \frac{\left( \frac{V_{out}^2}{R_{out}} \right)}{\left( \frac{V_{in}^2}{R_{in}} \right)} \right) = 10 \log_{10} \left( \frac{V_{out}^2}{V_{in}^2} \cdot \frac{R_{in}}{R_{out}} \right) \]

\[ G_{dB} = 20 \log_{10} \left( \frac{V_{out}}{V_{in}} \right) + 10 \log_{10} \left( \frac{R_{in}}{R_{out}} \right) \]
Decibel Scale

If input and output impedances are considered equal ($R_{in} = R_{out}$) the formula reduces to:

$$G_{dB} = 20 \log_{10} \left( \frac{V_{out}}{V_{in}} \right)$$

Linear gain is always positive (between 0 and $\infty$). Describe the gain in a dB range when the system is attenuating. Describe the gain in dB range when the linear gain is unity.
Bode Plot

- Bode plots provide information on the general behavior of circuits over a broad range of frequencies and magnitudes where important features exist.

- A TF Bode plot shows the magnitude in dB and phase in degrees over a frequency range and scale showing all important features of the plot.

- Important features of the plot include resonances, nulls, transitions between pass and stop bands, and asymptotic gains (not flat regions that change very little or in a predictable manner!).
Bode Plot (with Matlab)

- Before building our understanding on how to manipulate TF polynomials for convenient plotting, consider using Matlab to generate the plot from polynomials evaluated over a complex field.

- Matlab functions behave like a subroutine in a main program. Specify input and output arguments for your function (subroutine). Only the input and output arguments exist in the Matlab workspace or main program. Help information for the function is the first set of comments you include after the function definition. These comments must provide all the user needs to know to use it properly. At the Matlab prompt, typing >> help "function name" will display these comments.

- Example: Create a Matlab function that evaluates a TF based on its polynomial coefficients. Generate a script that calls this function to show how it works.
Matlab Example

- For a function, start the first line with the word “function” followed by the function syntax (input and output arguments). Comments immediately following this first line will be used as the “help” for this function and must show the complete syntax on how to use function along with descriptions of input and output variables. An example of how to call the function is helpful in the comments as well.

- First line of text file when creating a function

```matlab
function p = tfeval(ns, ds, f)
```
Matlab Example

The file should be saved with the same name as the function as defined in the first line. In this case it would be saved as "tfeval.m"

```
function p = tfeval(ns, ds, f)
% This function evaluates a transfer function for a given frequency(s)
% p = tfeval(ns, ds, f)
% where NS => the vector representing the numerator polynomial
% DS => the vector representing the denominator polynomial
% F => vector of frequency values in Hz at which the TF is to be evaluated
% P => vector of TF values corresponding to F.
% ns(1)*s^N + ns(2)*s^(N-1) + ... + ns(N)*s + ns(N+1)
% ds(1)*s^M + ds(2)*s^(M-1) + ... + ds(M)*s + ds(M+1)
```
% Example:
% for linear frequency evaluation
% >> freq = [0:10:10000];   % Frequency points
% >> numc = [10  0];   % Numerator coefficients
% >> denc = [1 5000*2*pi];   % Denominator coefficients
% >> tf = tfeval(numc, denc, freq); % Compute complex TF points
% >> figure; plot(f,abs(p)); ylabel('Magnitude') % Plot magnitude
% >> figure; plot(f,angle(p)*180/pi); ylabel('Degrees') % Plot Phase
%
% Updated by Kevin D. Donohue (donohue@engr.uky.edu) February 1, 2010

num_order = length(ns)-1;   % Determine order of numerator
den_order = length(ds)-1;   % Determine order of denominator

s = 2j*pi*f;   % Create s vector for evaluating TF
% Loop to sum up every term in numerator
sumn = zeros(size(s)); % Initialize accumulation variable
for k=1:num_order+1
   sumn = sumn + ns(k)*s.^(num_order-k+1);
end

% Loop to sum up every term in numerator
sumd = zeros(size(s)); % Initialize accumulation variable
for k=1:den_order+1
   sumd = sumd + ds(k)*s.^(den_order-k+1);
end

p = sumn ./ sumd; % Divide numerator and denominator points
Matlab Example Test Script

The following script will create the input variables for the function and plot the output.

% This script demonstrates the TFEVAL function by evaluating
% a transfer function at many points and plotting the result
% on both a linear and log scale
% Updated by Kevin D. Donohue (donohue@engr.uky.edu) 2/1/2010

% Generate 1000 points equally spaced on a Log Scale from
% 20 to 20kHz
f = logspace(log10(20), log10(20000), 1000);
% H(s) = -------------------------------
% s^2 + 1.81k*s + 900M
nu = [3 0]; % Numerator polynomial
de = [1, 1.81e3, 900e6]; % Denominator polynomial
tf = tfeval(nu, de, f); % Evaluate Transfer function
Matlab Example

Continued ...

figure(1)
% Plot result on a linear scale
plot(f,abs(tf)) % Plot magnitude
title('Transfer Function Example - Linear')
xlabel('Hertz')
ylabel('TF magnitude')

figure(2)
plot(f,phase(tf)*180/pi) % Plot phase
title('Transfer Function Example - Linear Scale')
xlabel('Hertz')
ylabel('TF phase (degrees)')
Matlab Example

Continued ...

figure(3)
%  Plot result on a log scale
semilogx(f,20*log10(abs(tf)))  %  Plot magnitude
title('Transfer Function Example - Log (dB) scale')
xlabel('Hertz')
ylabel('TF magnitude in dB')

figure(4)
semilogx(f,phase(tf)*180/pi)  %  Plot phase
title('Transfer Function Example - Log Scale')
xlabel('Hertz')
ylabel('TF phase (degrees)')
Plot Results

Linear Scales

Transfer Function Example - Linear Scale

TF magnitude

Hertz

TF phase (degrees)

Hertz
Plot Results

Bode Plots (Log Scales)

Transfer Function Example - Log (dB) scale

TF magnitude in dB

-140  -120  -100  -80  -60  -40
10^1  10^2  10^3  10^4  10^5  

Hertz

Transfer Function Example - Log Scale

TF phase (degrees)

-100  -50  0  50  100
10^1  10^2  10^3  10^4  10^5  

Hertz
Plot Ranges

The interesting parts of the plot occurs near the roots of the numerator (zeros) and denominator (poles). The only exception to this is roots at zero (which actually correspond to negative infinity on a log scale), which results in asymptotic behavior as \( \omega \) goes to zero or infinity on a log scale. So by doing an analysis of the roots of the TF, the plot range can be determined by starting a decade before the smallest root magnitude and going a decade after the largest root magnitude.

**Example:** Graph the magnitude and phase of the transfer function. Create both a linear scaled plot and a log (Bode) scaled plot.

\[
H(s) = \frac{\frac{4}{5} s^2}{s^3 + 6s^2 + \frac{14}{5} s + 4}
\]
Plot Ranges

The following Matlab script can be used to determine plot range:

```matlab
% Define range for w
% Find poles:
ps = roots([1, 6, (14/5), 4]); % Vector of polynomial coefficients
% ps will be a vector containing the poles
% Find zeros
zs = roots([(4/5), 0, 0]); % Vector containing polynomial coefficients, zs will be a vector containing the roots

% Find maximum magnitude pole and zero
fend1 = max(abs(ps)) % fend1 will be the maximum of the
% magnitudes of ps. (fend1 = 5.6288)
fend2 = max(abs(zs)) % fend2 will be the maximum of the
% magnitudes of zs. (fend2 = 0)
```
% Pick maximum value between the two (5.6 in this case), round up to the % next decade (which is 10) and increment to next decade (100 in this case)

% Find minimum magnitude pole and zero
fbeg1 = min(abs(ps))    % fbeg1 is the minimum of the magnitudes of ps.
%% (fbeg1 = 0.8430)
fbeg2 = min(abs(zs)    % fbeg2 is the minimum of the magnitudes of zs.
%% (fbeg2 = 0)
% Pick the minimum non-zero value between the two (.84 in this case), % round down to next decade (which .1) and decrement to next decade % (.01 in this case).

% Now create the w-axis vector with 201 equally spaced points % on a log (base 10) scale:
w = logspace(-2, 2, 201);    % w is now a vector of points from 10^-2 to 10^2 % Note that the transfer function is computed for s=j*w, therefore assign: s=j*w;    % now s is a vector of imaginary numbers (j=sqrt(-1) by default) % Now evaluate the transfer function at all points defined by s:
h = (4/5)*s.^2 ./ (s.^3 + 6*s.^2 + (14/5)*s + 4);
Plot Ranges

% Plot magnitude on semilog axis in decibels:
figure(1)
semilogx(w, 20*log10(abs(h)))
grid % Add gridlines to the plot
xlabel('Radians per Second') % Add x-axis label
ylabel('TF Magnitude in Decibels') % Add y-axis label

% Plot phase on semilog axis in degrees:
figure(2)
% angle return in radians, convert to phase.
semilogx(w, (180/pi)*(angle(h)))
grid % Add gridlines to the plot
xlabel('Radians per Second') % Add x-axis label
ylabel('TF Phase in Degrees') % Add y-axis label
Plot Results

Plot output (Log)

TF Magnitude in Decibels

TF Phase in Degrees

TF Magnitude in Decibels vs. Radians per Second

TF Phase in Degrees vs. Radians per Second
Plot Ranges

% For the linear plot redefine the range for w, the interesting part
% looks like it's between .1 and 20.
%   So create a new w-axis:
wl= [0:0.1:20];   % This creates and array of points starting at 0
                 % and goes up to 20 in increments of 0.1
s=j*wl;
h = (4/5)*s.^2 ./ (s.^3 + 6*s.^2 + (14/5)*s + 4);
%  Now  Create a linear plot for the magnitude:
figure(3)
plot(wl, abs(h))
grid         % Add gridlines to the plot
xlabel('Radians per Second') % Add x-axis label
ylabel('TF Magnitude')    % Add y-axis label
%  Now  Create a linear plot for the phase:
figure(4)
plot(wl, angle(h))
gird         % Add gridlines to the plot
xlabel('Radians per Second') % Add x-axis label
ylabel('TF Phase')    % Add y-axis label
Plot Results

Plot output (linear)
Complex Frequency

Since the poles and zeros of a transfer function can be complex (having a real and imaginary part), the \( s \) variable in the transfer function is referred to as complex frequency:

\[
s = \sigma + j\omega
\]

So a full plot of the TF is over the entire real and imaginary values in the complex number plane; however, for now we'll focus only on evaluating the TF on the \( j\omega \) axis (i.e. \( \sigma = 0 \)).
Bode Sketches

Bode plots are also useful for generating simple sketches of the transfer function due to the dB (logarithmic) scale, which converts multiplicative factors to additive terms.

Recall logarithmic relationship:

\[
\log \left| \frac{s - z_1}{(s - p_1)(s - p_2)} \right| = \log |s - z_1| - \log |s - p_1| - \log |s - p_2|
\]
Bode Sketches

Transfer function sketches can be performed by knowing the shape of 4 basic factors that any transfer function can be decomposed into:

- constant factors
- non-zero real pole or zeros factors
- pole or zero factors at zero
- complex conjugate pole or zero factors

Sketches are simplified by primarily considering the asymptotic behavior of each factor (i.e. as $\omega \to \infty$ and as $\omega \to 0$).

The phase and magnitude for each factor can be sketched individually and summed together in the final step.
Bode Sketches

Sketch the Bode plot of the transfer function:

Show for magnitude sketch in dB:

\[ \hat{H}(s) = \frac{10s(s + 100)}{s^2 + 10s + 100} \]

\[ 20 \log |\hat{H}(j\omega)| = 20 \log 10 + 20 \log |j\omega| + 20 \log \left| \frac{j\omega}{100} + 1 \right| - 20 \log \left| \frac{(j\omega)^2 + j\omega 10}{100} + 1 \right| \]

Show for phase sketch:

\[ \angle \hat{H}(j\omega) = \angle 10 + \angle j\omega + \angle \left( \frac{j\omega}{100} + 1 \right) - \angle \left( \frac{(j\omega)^2}{10} + \frac{j\omega 10}{100} + 1 \right) \]
Resonance

Resonance occurs when the capacitive and inductive reactance is equal in magnitude, leaving a purely resistive impedance. Resonance results in a local maximum (or minimum) point on the transfer function magnitude.

The Bandwidth ($B$) is the distance between the frequencies where the amplitude is down by a factor of $\sqrt{2}$ from the maximum. These frequencies are sometimes called the half power points.
The Quality Factor ($Q$) is the ratio of the resonant frequency to its bandwidth.

$$Q = \frac{\omega_o}{B}$$
Resonant Circuit Example

Find \( \omega_0 \), \( B \), \( Q \), and the gain at resonance \( (G_0) \) in terms of the circuit parameters where the input is \( v_i(t) \) and the output is \( i_o(t) \).

Show:
\[
\omega_0 = \frac{1}{\sqrt{LC}} \quad B = \frac{R}{L} \quad G_0 = \frac{1}{R} \quad Q = \frac{L \omega_0}{R}
\]

where \( (B \text{ and } \omega_0 \text{ are in Radians/Second}) \)

- Note: This formula set is only valid for this circuit. Similar formulae must be derived for different resonant circuits.
- In general a 2nd order resonant circuit can always be put in the form:
\[
\hat{H}(s) = \frac{G_0 Bs}{s^2 + Bs + \omega_0^2}
\]
Resonant Circuit Example

Find $\omega_0$, $B$, $Q$, and the gain at resonance ($G_0$) in terms of the circuit parameters where the input is $i(t)$ and the output is $v_o(t)$:

$$i(t) \quad R \quad C \quad L \quad v_o(t)$$

Show:

$$\omega_0 = \frac{1}{\sqrt{LC}} \quad B = \frac{1}{RC} \quad Q = R \sqrt{\frac{C}{L}} \quad G_0 = R$$

($B$ and $\omega_0$ are in Radians/Second)

- Note: This formula set is only valid for this circuit (Sometimes called a tank circuit). Similar formulae must be derived for different resonant circuits.